

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-212090

(43)Date of publication of application : 15.08.1997

---

(51)Int.Cl. G09C 1/00

H04L 9/14

---

(21)Application number : 08-039099 (71)Applicant : SONY CORP

(22)Date of filing : 02.02.1996 (72)Inventor : ENARI MASAHIKO

---

## (54) DECODING METHOD AND ELECTRONIC APPARATUS

### (57)Abstract:

PROBLEM TO BE SOLVED: To make it possible to complete all of decoding processing in real time on a reception side.

SOLUTION: PID information and TSC information are extracted from the inside of the header of the reception data inputted to a terminal 1 and are supplied to an IDT (inter-data key retriever) 16. This IDT 16 receives such information and searches and reads out data keys out of a DPMEM(dual port memory) 17 by an indirect retrieval method. The plural data keys updated at every prescribed period are written asynchronously in the DPMEM 17. The writing of the DPMEM 17 is prohibited when the writing and reading out of the DPMEM are simultaneously executed with the same address at the same timing. As a result the easy memory control of the DPMEM 17 is made possible.

<hr size=2 width="100%" align=center>

## CLAIMS

---

### [Claim(s)]

[Claim 1] A write-in step which writes two or more key information in a memory means and a read-out step which shifts and reads that key information from said memory means to be directed using information in input data. Consist of a decipherment step which decodes said input data based on key information read at this read-out step and A period of implementation of said write-in step. A decoding method which is a case where a period of implementation of said read-out step laps and is characterized by forbidding execution of said write-in step when a writing

address and a reading address are in agreement.

[Claim 2] A read-out step which is based on indicative data in input data shifts and reads that key information from a memory means which has memorized two or more key information. Consist of a decipherment step which decodes said input data based on key information read at this read-out step and in said read-out step. Key information read from said memory means by referring to a key table with said indicative data is searched. A decoding method choosing fewest key addresses when a key address for reading said key information applicable to said indicative data exists in said two or more key tables.

[Claim 3] By having the following and referring to a table with said indicative data in said read-out step. A decoding method having searched a key address of key information read from said memory means and searching a key address when a key address referred to with said indicative data does not exist in said table.

A read-out step which is based on indicative data in input data shifts and reads that key information from a memory means which has memorized two or more key information.

A decipherment step which decodes said input data based on key information read at this read-out step.

[Claim 4] By having the following and referring to a key table with said indicative data in said read-out step. A decoding method characterized by carrying out initial processing in said initial step so that there may be no applicable key address as a result of searching a key address of key information read from said memory means and searching within an initialization period.

An initial step which performs initial setting.

A write-in step which writes two or more key information in a memory means.

A read-out step which is based on indicative data in input data shifts and reads that key information from said memory means.

A decipherment step which decodes said input data based on key information read at this read-out step.

[Claim 5] Electronic equipment provided with a decoding means which performs a decoding method according to claim 1 at least.

[Claim 6] Electronic equipment provided with a decoding means which performs a decoding method according to claim 2 at least.

[Claim 7] Electronic equipment provided with a decoding means which performs a decoding method according to claim 3 at least.

[Claim 8] Electronic equipment provided with a decoding means which performs a decoding method according to claim 4 at least.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention]It is enciphered in this invention.

Thereforein response to the send data by which scramble was carried outit is related with the decoding method and electronic equipment which were decoded.

[0002]

[Description of the Prior Art]In order to keep the information in communication secretthe encryption and the decoding method which acquired the original information are known from the former by enciphering transmit informationand receiving and decoding the enciphered transmit information. Cryptographic algorithmssuch as DES (Data Encryption Standard) which is a standard method in the U.S. as such encryption and a decoding methodare known.

[0003]By the waythere are a variety and various things in a cryptographic algorithmand the method which was more excellent in safety and rapidity is developed. As this examplethe cipher system (MULTI2 method) indicated to the US4982429B Descriptionthe US5103479B DescriptionJPH1-276189Aetc. is known. There are a cipher system registered as ISO9979/0009 also in International Organization for Standardization (ISO) and encryption use mode registered as ISO/IEC10116.

[0004]In the cipher system of the MULTI2 above-mentioned methodinput data size shall be 64 bitsoutput data size is 64 bitsand the work key of 256 bit sizes required in order to perform encryption is generated from the system key of 256 bit sizeand a 64-bit data key. Let the encryption number of stages be a positive integer stage. The outline composition of the encryption algorithm in this MULTI2 method is shown in drawing 17. MULTI2 method generates the 256-bit work key Kw by using 256-bit system key J for the 64-bit data key Ksand calculating a cryptographic algorithmas shown in drawing 17. The operation of this cryptographic algorithm is performed by the cryptographic algorithm execution means C. The 64-bit plaintext which the generated work key Kw was supplied to the cryptographic algorithm execution means Fand was inputted is enciphered. The cryptographic algorithm performed by the cryptographic algorithm execution means C and the cryptographic algorithm execution means F is the same cryptographic algorithm.

[0005]Although such encryption is a fundamental encryption algorithm of MULTI2 methodNowthere is a possibility that a plaintext may be presumedby carrying out the statistical work of the distribution of the frequency where a character or a word appears beforehandand taking matching with the frequency distribution of the character string pattern of the encryption sentence which came to hand. Thenthere is the technique of calculating the exclusive OR of the enciphered 64 bits code block and the 64-bit input data inputted into the nextand creating a cryptogram. The mode

which performs this technique and is enciphered is read with the CBC (Cipher Block Chaining) mode. The cryptographic algorithm in such CBC mode is performed in the above mentioned cryptographic algorithm execution means F.

[0006] Although there is a communication method with which the unit of the data which communicates for example like packet communication is decided beforehand when the data unit which cannot be divided among the number of bits of 1 block is inputted data comes to remain in a block cipher system which makes 64 bits 1 block. Then he is trying to process the fraction process in OFB (Output Feedback) mode. In this OFB mode the fractional part of data is supplied to the cryptographic algorithm execution means G and is enciphered using a random number. This random number is generated by the cryptographic algorithm execution means G using the work key Kw. Thereby the cryptogram which makes 64 bits 1 block can be acquired now. The CBC mode and OFB mode are called encryption use mode.

[0007] The outline composition of the decoding algorithm in MULTI2 method is shown in drawing 18. As shown in drawing 18 the 256-bit work key Kw is generated by using 256-bit system key J for the 64-bit data key Ks and calculating a cryptographic algorithm. The operation of this cryptographic algorithm is performed by the cryptographic algorithm execution means c. Decoding of the 64-bit cryptogram which the generated work key Kw was supplied to the decipherment algorithm execution means f and was inputted is carried out. The cryptogram enciphered in OFB mode is supplied to the cryptographic algorithm execution means g and decoding is carried out by using the random number generated by the cryptographic algorithm execution means g using the work key Kw. Thereby decoding of the 1-block 64-bit cryptogram can be carried out and a 64-bit plaintext can be obtained. When considered as the CBC mode it is made for the decipherment algorithm execution means f to have a decipherment algorithm in the CBC mode performed.

[0008] Here although explanation of encryption use mode is given referring to drawing 19 the outline composition of encryption and decoding of the CBC mode is shown in drawing 19 (a) and the outline composition of encryption and decoding of OFB mode is shown in drawing 19 (b). In the CBC mode as shown in drawing 19 (a) i-th plaintext block M(i) is inputted into the exclusive "or" circuit 101 and exclusive OR with ciphertext block C (i-1) in front of 1 block delayed with the register (REG) 103 calculates it. The calculated data is enciphered in the cryptographic algorithm execution means 102 by the work key generated based on the data key Ks. This enciphered i-th ciphertext block C(i) is  $C(i) = E_{Ks}(M(i) \text{ .EOR. } C(i-1))$ .

It can express. However it means that  $E_{Ks}(m)$  enciphers m by Ks and it is shown that EOR calculates exclusive OR.

[0009] And it will be transmitted and this ciphertext block C(i) will be received in a receiver. Received ciphertext block C(i) is decoded using the work key generated based on the data key Ks in the decipherment algorithm execution means 111 and is supplied to the exclusive \*\*\*\* circuit 113. Ciphertext block C (i-1) in front of 1 block

delayed in the register (REG) 112 is inputted into this exclusive "or" circuit 113 and both exclusive \*\*\*\* calculates. At this time the data key Ks of the transmitting side and a receiver is equal and thereby i-th plaintext block M(i) is decoded from the exclusive "or" circuit 113. i-th plaintext block M(i) can be expressed as follows.

$$M(i) = DKs(C(i) .EOR.C(i-1))$$

However it is shown that DKs (c) carries out decoding of the c by Ks.

[0010] In the time of OFB mode i-th plaintext block M(i) is supplied to the exclusive "or" circuit 105. The output of the cryptographic algorithm execution means 104 random-number-ized by the work key generated based on the data key Ks is supplied to this exclusive "or" circuit 105. 1 block of outputs of the cryptographic algorithm execution means 104 are delayed with the register 103 and are returned to the cryptographic algorithm execution means 104. Thereby from the exclusive "or" circuit 105 ciphertext block C(i) enciphered with the random number is outputted.

[0011] And it will be transmitted and this ciphertext block C(i) will be received in a receiver. Received ciphertext block C(i) is supplied to the exclusive \*\*\*\* circuit 114. The output random-number-ized using the work key generated based on the data key Ks in the cryptographic algorithm execution means 115 is supplied to this exclusive \*\*\*\* circuit 114. 1 block of outputs of this cryptographic algorithm execution means 115 are delayed in the register (REG) 112 and are returned to the cryptographic algorithm execution means 115. In this case the random number supplied to the exclusive "or" circuit 114 is equal to the random number supplied to the exclusive "or" circuit 105 and thereby i-th plaintext block M(i) is obtained from the exclusive "or" circuit 114.

[0012] The outline composition of the encryption and the decoding method which has the encryption use mode explained above is shown in drawing 20. In this figure the transmitting side is equipped with the scrambler 100 and the scramble of the input data is carried out by the scrambler 100 and it is transmitted. The transmission line of space etc. is spread by this send data by which scramble was carried out and it is received by a receiver. The receiver is equipped with the descrambler 110.

The send data scramble was carried out [ send data ] by this descrambler 110 is descrambled and it comes to be returned and outputted to the original data.

[0013] Encryptor 102 which is a cryptographic algorithm execution means which enciphers the input data (plaintext) into which the scrambler 100 was inputted. It comprises an OFB mode encryption section set to the register 103, the CBC mode encryption section which consists of the exclusive "or" circuit (EX-OR) 101 and Encryptor 104 which are cryptographic algorithm execution means from the exclusive "or" circuit (EX-OR) 105. It has Encryptor 106 which generates a work key from a data key and a system key in the scrambler 100. The generated work key is supplied to Encryptor 102, 104. By the way, since Encryptor 102, Encryptor 104 and Encryptor 106 are made into the identical configuration, three Encryptor(s) can be made to serve a

double purpose by one Encryptor. Since operation of the CBC mode encryption section and an OFB mode encryption section is as having mentioned above it is omitted here.

[0014] Decryptor 111 which is a decipherment algorithm execution means by which the descrambler 110 carries out decoding of the inputted received data (cryptogram) It comprises an OFB mode decoding part set to the register 112 the CBC mode decoding part which consists of the exclusive "or" circuit (EX-OR) 113 and Encryptor 115 which are cryptographic algorithm execution means from the exclusive "or" circuit (EX-OR) 114. It has Encryptor 116 which generates a work key from a data key and a system key in the descrambler 110. The generated work key is supplied to Decryptor 111 and Encryptor 115. Since Encryptor 115 and Encryptor 116 are made into the identical configuration two Encryptor(s) can be made to serve a double purpose by one Encryptor. Since operation of the CBC mode decoding part and an OFB mode decoding part is as having mentioned above it is omitted here.

[0015]

[Problem(s) to be Solved by the Invention] By the way when realizing and using advanced encryption and decoding method like MULTI2 method mentioned above by software a real-time operation cannot be performed in the computing speed of the present calculating means. that is when it applies for example to satellite digital television broadcast etc. decoding processing [ in / for a picture or a sound / a receiver since / without \*\*\*\*\* / it is necessary to come out and carry out real-time reproduction ] must be performed by the possible hardware of high-speed processing. However the hardware which can complete all the decipherment processings in real time in a receiver had the problem of becoming large-sized from performing complicated processing.

[0016] Then this invention sets it as the 1st purpose to provide the decoding method which can complete all the decipherment processings for the data by which scramble was carried out with the advanced cipher system using small and cheap hardware in a receiver in real time. This invention sets it as the 2nd purpose to provide small and cheap electronic equipment provided with the decoding means which can complete all the decipherment processings for the data by which scramble was carried out with the advanced cipher system in real time in a receiver.

[0017]

[Means for Solving the Problem] In order to attain the 1st purpose of the above a decoding method of this invention A write-in step which writes two or more key information in a memory means and a read-out step which shifts and reads that key information from said memory means to be directed using information in input data Consist of a decipherment step which decodes said input data based on key information read at this read-out step and A period of implementation of said write-in step It is a case where a period of implementation of said read-out step laps and when a writing address and a reading address are in agreement the is trying to forbid

execution of said write-in step.

[0018]In order to attain the 1st purpose of the aboveother decoding methods of this inventionA read-out step which is based on indicative data in input datashiftsand reads that key information from a memory means which has memorized two or more key informationConsist of a decipherment step which decodes said input data based on key information read at this read-out stepand in said read-out step. Key information read from said memory means by referring to a key table with said indicative data is searchedWhen a key address for reading said key information applicable to said indicative data exists in said two or more key tableshe is trying to choose fewest key addresses.

[0019]In order to attain the 1st purpose of the abovefurther again a decoding method of further others of this inventionA read-out step which is based on indicative data in input datashiftsand reads that key information from a memory means which has memorized two or more key informationConsist of a decipherment step which decodes said input data based on key information read at this read-out stepand in said read-out step. A key address of key information read from said memory means by referring to a table with said indicative data is searchedand when a key address referred to with said indicative data does not exist in said key tableit is made not to search a key address.

[0020]In order to attain the 1st purpose of the abovefurther again a decoding method of further others of this inventionAn initial step which performs initial settingand a write-in step which writes two or more key information in a memory meansA read-out step which is based on indicative data in input datashiftsand reads that key information from said memory meansConsist of a decipherment step which decodes said input data based on key information read at this read-out stepand in said read-out step. A key address of key information read from said memory means by referring to a key table with said indicative data is searchedand as a result of searchingin said initial stepinitial processing is made to be carried out within an initialization period so that there may be no applicable key address.

[0021]He is trying to provide at least electronic equipment of this invention which attains the 2nd purpose of the above with a decoding means which shifts and performs that decoding method to describe above further again.

[0022]Since read-out and writing control of a memory means in which key information required in order to perform decipherment processing is stored can be performed appropriately according to a decoding method of such this inventionit can be considered as a decoding method that memory control can be performed easily. Thereforeit can be considered as a decoding method that all the decipherment processings can be performed in real time in a receiver. In electronic equipment provided with a decoding means which performs such a decoding methodsince composition of a memory control means can be simplifiedelectronic equipment can be provided small and cheaply.

[0023]

[Embodiment of the Invention] The block diagram showing the example of composition of the decipherment device which is an embodiment of the decoding method of this invention is shown in drawing 1. In this figure the decipherment device A The decoding part B in the CBC mode and the decoding part C in OFB mode. The data key indirect search machine (IDT) 16 provided with a packet ID (PID) table It comprises the comparator (COMP) 11 which compares with the writing address and reading address of DPMEM 17 the dual port memory (DPMEM) 17 which passes the data key used for encryption of input data to the decoding parts B and C.

[0024] The cryptogram data which is received data is inputted into the terminal 1 of the decipherment device A and this cryptogram data is inputted into the means for switching 3. When [ both ] considered as the CBC mode at this time the means for switching 3 and the means for switching 4 are switched to the terminal a side decoding of the inputted cryptogram data is supplied and carried out to the CBC mode decoding part B and it is outputted to it. When considered as OFB mode it is switched to the means for switching 3 both the means for switching 4 are switched to the terminal b side and cryptogram data is inputted into the OFB mode decoding part C side. And decoding of the inputted cryptogram data is carried out in the OFB mode decoding part C and plaintext data is outputted. In the CBC mode decoding part B and the OFB mode decoding part C although decipherment processing of the inputted cryptogram data is carried out and it is considered as plaintext data the decoding algorithm is the same as the decoding algorithm shown in said drawing 18. A data key required for decipherment processing is supplied from DPMEM 17 in this case.

[0025] Let the format of the received data inputted into the terminal 1 be the transport stream (it is hereafter described as TS.) specified for example as ISO/IEC 13818. TS is made into 188 bytes of packet structure and he is trying for the pay load of 184 bytes to usually continue after 4 bytes of header. Since parity is added for the error correction to a transmission error 16 bytes of straw-man period for parity is added and it is considered as the stream of these repetitions.

[0026]. [ whether the packet is constituted from picture image data by the header of this TS and ] The PID information which shows whether it comprises comprising voice data or other data The TSC (Transport Scrambling Control) flag etc. which show whether it is TS enciphered are contained and the attribute of these packets is interpreted by analyzing a header in the analyzing parts which are not illustrated. It is made to output TS as it is through a delay part when interpreted as the packet which is not enciphered by TSC at this time without performing decoding processing. The time delay of this delay part is made equal to the time which decipherment processing takes to the decipherment device A.

[0027] To one PID two the data key  $K_{se}$  ( $K_{s\_even}$ ) and the data key  $K_{so}$  ( $K_{s\_odd}$ ) are assigned. Since a data key is updated every [ several seconds thru/or ] tens of seconds this needs to rewrite a data key so that it may be enough for updating And it



is because only one of these is used and updating of the data key which is not used of the data key Kse and the data key Kso is enabled and they can update the data key at the time of decipherment processing. That is it is for performing easily updating control which updates a data key. Although said analyzing parts do not illustrate it prepares for the CBC mode decoding part B for example and PID/TSC information with a data size of 15 bits is sent to IDT16 from the CBC mode decoding part B.

[0028] In IDT16 read address RA (data size is 9 bits) of DPMEM17 is generated combining the address information read from PID with reference to the PID table using the PID information of 13 received bit sizes and the TSC information on 2 bit size. Generated read address RA was supplied to RA terminal of DPMEM17 from AO terminal it was controlled so that a lead enabling (RE) signal would be in an active state at this time and it has read the data key applicable to read address RA from DPMEM17. The read data key is equal to the data key used at the time of encryption of inputted TS is outputted from terminal RD and supplied to Decryptor 5. Decryptor 5 performs decipherment processing of the cryptogram data which generated the work key and was inputted based on the generated work key from the supplied data key and a different system key for every system. The system key is beforehand passed to the decipherment device A from CPU10.

[0029] When you are trying to be written in DPMEM17 and it writes in the data key which is needed in the decoding parts B and C and which is updated before CPU10 is needed in decipherment processing Write enable (WE) is made into an active state and the write data (WD) of a write address (WA) with a data size of 9 bits and a data key with a data size of 8 bits is supplied to DPMEM17. In this case CPU10 is equipped with ROM or RAM and the data given to the decipherment device A is written in these memories. Since the read data becomes unfixed when writing data key information in DPMEM17 and IDT16 comes to read a data key with the same read address as a write address the writing of DPMEM17 has been forbidden at this time.

[0030] For this reason the comparator 11 AND gate 12 OR gates 13 and 14 and the inverter 15 are formed. Explanation of this operation will input into the relay terminal and Q terminal of the comparator 11 read address RA outputted from IDT16 and the write address WA outputted from CPU10 respectively. And when the input data of a relay terminal and the input data of Q terminal are in agreement ( $P=Q$ ) a high level signal is outputted from the comparator 11. Since high level will be outputted from the inverter 15 if the reversal RE is an active state with a low level at this time high level will be outputted from AND gate 12 despite a join office. Since the output of this AND gate 12 is made are inputted into OR gate 13 and high-level [ the output of OR gate 13 ] the reversal WE is inactive and the writing of DPMEM17 will be forbidden.

[0031] Next although the memory space of the register of the decipherment device A seen from CPU10 is shown in drawing 3 The area of 64bit x1 where top 6 bits of an address (HADD) with a data size of 9 bits were set to "000100" is assigned to the initial value table (CBC Initial value table) in the CBC mode. At the time of the initial

processing performed by the power up etc. the initial value memorized by this initial value table is supplied to the CBC mode decoding part Band is set in the register 6. The area of 256bit x1 where top 4 bits of an address (HADD) were set to "0010" is assigned to the system key table (SYSTEM\_Key table). Although these system keys shall differ for every system it is the key fixed in one system.

[0032] The area of 13bit x12 where top 4 bits of an address (HADD) were set to "0100" is assigned to the PID table (PID value table). This PID table is information which shows the packet-sized kind of data and it is referred to as PID which is made into a maximum of 12 kinds and differs for every kind of data in one channel. The information on a PID table is written in a register predetermined in a system key table from CPU10 IDT16 at the time of initial processing.

[0033] The area of 64bit x12 where top 2 bits of an address (HADD) were set to "10" further again it is assigned to a data key Kse table (Ks\_even value table) and the area of 64bit x12 where top 2 bits of an address (HADD) were set to "11" is assigned to the data key Kso table (Ks\_odd value table). Although the information on this data key Kse table and a data key Kso table is written in DPMEM17 it is updated by CPU10 for every prescribed timing for several seconds thru/or several 10 seconds.

[0034] Next the block diagram showing an example of the detailed composition of IDT16 is shown in drawing 2. In this figure 41-52 are 12 flip-flops (DF0-DF11) which have 13 bit width respectively and 61-72 are 12 comparators (CP0-CP11) which have 13 bit width respectively. 40 is an address decoder (ADEC) and if address information with a data size of 9 bits is inputted the output which activates either DF0-DF11 will be taken out only to one in the output of 12. 39 is a priority encoder (PE) and the address information TMP0-TMP3 combined with TSC information is outputted.

[0035] In this IDT16 it is developed by the data size of 13 bits and PID information with a data size of 8 bits outputted from CPU10 is inputted into the terminal 30. Address information with a data size of 9 bits outputted from CPU10 is inputted into the terminal 32 and it is decoded in ADEC40. This decode output is inputted into DF0-DF11 as a strobe signal and the latch of only any one of DF0-DF11 of it is enabled. At this time PID information with a data size of 13 bits inputted is inputted common to 12 DF0-DF11 DF0-one every DF 11 are chosen one by one by the decode output of ADEC40 and the PID information currently supplied is latched. Thus a maximum of 12 kinds of PID information can be latched to DF0-DF11 and a PID table comes to be constituted by DF0-DF11.

[0036] PID information with a data size of 13 bits is inputted into the terminal 33 and it is inputted into it common to B input terminal of the comparators CP0-CP11. This PID information is PID information with a data size [ of the PID/TSC information with a data size of 15 bits which was extracted from the header of the packet which is received data and was supplied to IDT16 ] of 13 bits in the CBC mode decoding part B. In this case the PID information from DF0-DF11 is inputted into the generator terminal at the comparators CP0-CP11 respectively and a coincidence signal is outputted from

the comparator as which the PID information inputted from the terminal 33 and PID information in agreement are inputted into the generator terminal. Therefore if the PID information in a PID table is inputted from the terminal 33 a coincidence signal will be outputted from any one of CP0–the CP11.

[0037] Although this logical table of CP0–CP11 is shown in drawing 4 signal is outputted when in agreement [ all “1” is not inputted into the generator terminal and battery terminal of CP0–CP11 and ] (A=B). “0” is outputted when not in agreement. The logical table of PE39 is shown in drawing 5. In the logical table of this PE39 when CP2 to “1” signal is outputted from PE39 the 4 bits (TMP0–TMP3) data of “0010” is outputted. In this PE39 as shown in drawing 5 D0 is considered as the input to which priority was given most and D11 is considered as the input of the lowest priority. By the way although the PID information of the PID table set as DF0–DF11 and conflicting PID information may be inputted all “0” will be inputted into PE in this case. NK output of PE is set to “1” at this time. When NK output is “1” going to read the key table of the data key mentioned later is forbidden.

[0038] Since the level of the terminal 31 is set to “0” at the time of initial processing such as a power up and all of DF0–DF11 are preset DF0–DF11 to “1” is outputted. Therefore CP0–CP11 to all “0” will be outputted in this case and going to read the key table of a data key as mentioned above is forbidden. Since a key table is not used when going to read a key table is forbidden a data key can be written in this key table from CPU10. That is initial setting of a data key can be carried out. The terminal 31 is a power-on-reset terminal and the level returns after predetermined time “1.”

[0039] Next the operation which searches the key table of a data key is explained using the PID information and the TSC information in received data referring to drawing 6. Suppose that “PID–F” was inputted as PID information with a data size of 13 bits from the header of received data and “10” was inputted as TSC information with a data size of 2 bits at Step S20 shown in drawing 6. This “10” assumes that it is shown that scramble is carried out with the data key Kse. Subsequently this PID information is inputted into the generator terminal of CP0–CP11 at Step S21 and it is compared with the PID information stored in DF0–DF11. As a result CP5 to “1” signal is outputted and output TMP of TEMP3–TEMP0 of PE39 [3 . 0] is set to “0101.” That is PID–F is searched. “0101” which is an output of this PE39 is made into the 1st bit from the 4th bit of the address HADD.

[0040] Subsequently 2 bits (“10”) of TSC information TSC [1..0] are made into the 5th bit of HADD and the 6th bit at Step S22 and the address HADD of 6 bit size [8..3] is generated. Therefore the address HADD [8..3] becomes “100101.” And if the address HADD of “100101” generated at Step S23 [8..3] refers a key table data key Kse–F of 64 bit width will come to be obtained from Ks\_even Table. And the decipherment of a cryptogram inputted based on obtained data key Kse–F is performed with the decipherment device A. Thus without preparing the table of 13 bit width corresponding

to PID information since it is considered as the indirect search method which searches a data key with the decoding method of this invention indirectly since what is necessary is just to prepare the key table of 6 bit width the capacity of a memory can be reduced and it can be miniaturized.

[0041] In this way in a decipherment since only one side of Ks\_even Table of a key table and Ks\_odd Table is used the data key of the key table which is not used can be updated. As mentioned above CPU10 performs this but since the timing of CPU10 and the timing of the decipherment device A are asynchronous and it operates CPU10 goes the data key which it is asynchronous and is updated to write in. For this reason in the dual port memory 17 in which a key table is stored the case where writing and read-out are simultaneously performed by the same address as mentioned above arises. Although the example of timing at this time is shown in drawing 9 when the reversal RE of 8 clock width occurs to the timing to illustrate as mentioned above the writing of the same address from CPU10 is forbidden in these 8 clock periods.

[0042] By the way the case where the same PID is stored in the PID table by a certain cause arises. For example although considered as a maximum of 12 kinds per channel of PID information when you do not need 12 kinds of PID it is not necessary to write 12 kinds of PID in a PID table and only PID of the kind which carries out necessity will be written in it. Then the data of the column of PID which is not written in may turn into the data same by chance as PID. In such a case since there is a possibility that it may become impossible to read and decode the mistaken data key in this invention this has been prevented as follows.

[0043] As shown in drawing 7 when two or more "PID-F" is shown in a PID table he is trying for the address HADD of them to give priority to the smaller one. This is because it goes to write in a PID table from the smaller one of the address HADD so the probability which the larger one of the address HADD has mistaken is high. By this if "PID-F" is inputted as PID information 01000101 will come to be obtained as the address HADD [8..1] Processing mentioned above and same processing are performed and data key Kse-F of 64 bit width comes to be obtained from Ks\_even Table.

[0044] There may be no PID applicable by the PID information from the header of input data even if it refers to a PID table. For example as shown in drawing 8 even if "PID-F" is inputted as PID information there is no PID applicable to "PID-F" all over a PID table. In such a case a data key is not read without going to read both Ks\_even Table of a key table and Ks\_odd Table. In this case TS which is input data will be outputted as it is.

[0045] Next although the decipherment flow of the decipherment device A shown in drawing 1 is shown in drawing 10 if a transport stream (TS) is inputted it will be judged whether scramble is carried out at Step S10. This judgment is judged by detecting whether the flag in the header which shows that scramble was carried out stands. In this case when the flag stands it is judged with the scramble on and it progresses to Step S11 and rewriting of a desired flag etc. is performed here. Subsequently PID

information is extracted from a header at Step S12 and a key table is referred to. The key table in which host interface processing was performed at Step S16 of key table processing and the key table in this case was written in at Step S17 is referred to. Processing of Step S12 is processing shown in said drawing 6.

[0046] Processing of the above step S10 thru/or Step S12 is header control processing. When judged with the scramble off at Step S10 TS is outputted as it is. Subsequently decipherment processing in the CBC mode is performed at Step S13 and it is judged at Step S14 whether the cryptogram by which decipherment processing is carried out is an integral multiple which is 64 bits. When a cryptogram has a fraction and it is judged with No decipherment processing in OFB mode is performed about a fractional part at Step S15 and the plaintext by which decipherment processing was carried out is outputted. When judged with a 64-bit integral multiple at Step S14 the 64-bit plaintext by which decipherment processing was carried out comes to be outputted.

[0047] By the way execution of the decipherment flow as shown in drawing 10 is performed by the decoding algorithm shown in said drawing 18. Since the decoding algorithm shown in drawing 18 is as having mentioned above it omits here but the details of composition of performing the decipherment algorithm and cryptographic algorithm in a decoding algorithm are explained with reference to drawing 11 thru/or drawing 15. Drawing 11 shows the composition of cipher processing which performs a cryptographic algorithm. In drawing 11 the input data of 64 bit width is divided into top 32 bits data and 32-bit low-ranking data and is inputted into the first eight steps of codes. Eight steps of this code is considered as the composition by which four steps of operation stages which calculate a function were repeated twice. And in the first rank of the operation stage 20 the operation of the function  $\pi_1$  is performed to top 32 bits data and the 32-bit low-ranking data which were inputted. Subsequently in the 2nd step the operation of the function  $\pi_2$  is performed to the output of the first rank. In this case the work key K1 of 32 bit width is inputted into the 2nd step and the 2nd-step operation is performed using this work key K1.

[0048] In the 3rd step the operation of the function  $\pi_3$  is performed to the 2nd-step output. In this case the work key K2 of 32 bit width and K3 are inputted into the 3rd step and the operation is performed using this work key K2 and K3. Then in the 4th step the operation of the function  $\pi_4$  is performed to the 3rd-step output. In this case the work key K4 of 32 bit width is inputted into the 4th step and the operation is performed using this work key K4. In the operation stage 21 which performs four steps of operations which furthermore remain the operation of the function  $\pi_1$  is performed to the output from the operation stage 20 in the first rank. Subsequently in the 2nd step the operation of the function  $\pi_2$  is performed to the output of the first rank. In this case work key K5 of 32 bit width is inputted into the 2nd step and the operation is performed using this work key K5.

[0049] In the 3rd step the operation of the function  $\pi_3$  is performed to the 2nd-step

output. In this case the work key K6 of 32 bit width and K7 are inputted into the 3rd step and the operation is performed using this work key K6 and K7. Then in the 4th step the operation of the function pi 4 is performed to the 3rd-step output. In this case the work key K8 of 32 bit width is inputted into the 4th step and the operation is performed using this work key K8. Thus as for eight steps of code the data of a total of 64 bit width of top 32 bits and 32 bits of low ranks in which cipher processing was performed is further inputted into 22. As for eight steps of this code the operation of eight steps of codes mentioned above in 22 and the same operation are performed and the output data by which top 32 bits and a total of 64 bit width of 32 bits of low ranks were randomized is obtained.

[0050] The number of cycles of eight steps of codes is [ number of times of not only 2 times but a request ] repeatable as illustrated. Output data is randomized highly and can make encryption strength a strong thing so that many this number of times is repeated. the operation of the function currently performed by the operation stage coming out markedly is done to the substitution which transposes the character according to a fixed rule to other characters and the operation which performs the transposition which replaces an order of a character.

[0051] Next although the composition of the decipherment processing which performs a decipherment algorithm is shown in drawing 12a different point from cipher processing mentioned above is a point of being made to calculate conversely from the output side of the composition of cipher processing. That is in four steps of operation stages 23 of the beginning of eight steps of code the work key K8 of 32 bit width was used for the input data in which 64 bit width divided into top 32 bits and 32 bits of low ranks is enciphered and the function pi 4 is calculated. Subsequently in the 2nd step the work key K7 was used for the output data of the first rank and the function pi 3 is calculated. In the 3rd step the work key K6 and K7 were used for the output data of the 2nd step and the function pi 2 is calculated. In the 3rd step work key K5 was used for the output data of the 2nd step the function pi 2 was calculated and the function pi 1 is calculated to the output data of the 3rd step in the 4th step further again.

[0052] Four steps of such operations are similarly performed using the work keys K4-K1 in the operation 24. Also in 25 eight steps of codes in which the operation of eight steps of above-mentioned codes is concatenated performed and the output data of a total of 64 bit width of top 32 bits and 32 bits of low ranks by which decoding was carried out comes to be obtained. Let repeat frequency of eight steps of codes be a repeat count of eight steps of codes performed in cipher processing and the same number of times.

[0053] Next it explains in detail raising the operation stage 20 of cipher processing for the details of the operation currently performed in the operation stage to an example and referring to drawing 13. In the operation of the function pi 1 of the first rank the high order bit divided into inputted 32 bits is outputted as it is without calculating and the exclusive OR of a high order bit and a lower bit calculates it and it

is outputted as a lower bit. In the operation of the continuing function pi 2 of the 2nd step the work key K1 is added to the data x of 32 bits of low ranks and  $x+K1$  calculates first. Subsequently when  $x+K1$  is set to y the 1-bit left cyclic shift of the y is carried out  $y-1$  is added to the value and z is obtained. Next the 4-bit left cyclic shift of the z is carried out and the exclusive OR of the value and z is acquired. This result of an operation and top 32-bit exclusive OR calculate and calculated top 32-bit data is outputted. In this case 32 bits of low ranks are outputted as it is without the inputted data calculating.

[0054] In the operation of the function pi 3 of the 3rd step the work key K2 is added to top 32-bit data x and  $x+K2$  calculates first. Subsequently when  $x+K2$  is set to y the 2-bit left cyclic shift of the y is carried out  $y+1$  is added to the value and z is obtained. Next the 8-bit left cyclic shift of the z is carried out and the exclusive OR a of the value and z is acquired. The work key K3 is added to a and  $a+K3$  calculates. Subsequently when  $a+K3$  is set to b the 1-bit left cyclic shift of the b is carried out  $b$  is added to the value and c is obtained. Next the logical sum for every bit of a and x and exclusive OR with the value which carried out the 16-bit left cyclic shift of the c are calculated. The exclusive OR of this result of an operation and the data of 32 bits of low ranks is calculated and the data of 32 bits of calculated low ranks is outputted. Top 32-bit data turns into top 32-bit output data as it is without calculating.

[0055] By the operation of the function pi 4 of the 4th step the work key K4 is added to the data x of 32 bits of low ranks and  $x+K4$  calculates first further again. Subsequently when  $x+K4$  is set to y the 2-bit left cyclic shift of the y is carried out and  $y+1$  is added to the value. This result of an operation and top 32-bit exclusive OR calculate and calculated top 32-bit data is outputted. In this case the data of 32 bits of low ranks is outputted as data of 32 bits of low ranks as it is without calculating.

[0056] In the above-mentioned operation by adding the work keys K1-K4 to data substitution processing which replaces a character in other characters is performed and the transposition which replaces the position of a character is performed by carrying out the cyclic shift of the data. Thus a plaintext is enciphered by the cryptogram by performing substitution and the algorithm of transposition. When carrying out decoding it can decode to the original plaintext by performing substitution contrary to encryption and the algorithm of transposition.

[0057] Next although the composition which calculates the function mentioned above is explained still in detail the example of the function pi 2 shall be raised to drawing 14 and shall be explained. In drawing 14 in 1st adding machine 80 input data x and the 32-bit work key K1 of 32 bits of low ranks are added and addition data y is outputted. In the 1st left round shifter 81 the 1-bit left cyclic shift of this addition data y is carried out and it is added with the output of the 1st left round shifter 81 in the 2nd adding machine 82. In the 3rd adding machine 84  $-1$  is added to this added result and the data z calculates. In the 2nd left round shifter 85 the 4-bit left cyclic shift of this data z is carried out and it is supplied to the exclusive "or" circuit 86. The

output data of the 2nd left round shifter 85 the data z and top 32 bit input data are inputted into this exclusive "or" circuit 86 and the exclusive OR of three data calculates. This result of an operation serves as top 32 bit input data inputted into the next step. Low rank 32 bit input data turns into low rank 32 bit input data inputted into the next step without calculating.

[0058] Next the composition of the key schedule processing which generates the work key of 256 bit width from the data key of 64 bit width and the system key of 256 bit width is shown in drawing 15. Key schedule processing is considered as the composition to which one-step cascade connection of two steps and one step of operation stage 28 was carried out for four steps of operation stages 26 and 27 as shown in drawing 15. In four steps of operation stages 26 and 27 the operation of the function pi 1 is performed in the first rank the operation of the function pi 2 is performed in the 2nd step the operation of the function pi 3 is performed in the 3rd step and the operation of the function pi 4 is performed in the 4th step.

[0059] Since such an operation algorithm is the same as the algorithm of cipher processing mentioned above omit the explanation but. In key schedule processing input data is used as a 64-bit data key the operation of the function pi 1 thru/or the function pi 4 is performed using the 32-bit system keys J1-J8 respectively and the eight 32-bit work keys K1-K8 are generated respectively. However nine steps of operations are performed on the whole and it is different from the algorithm of cipher processing mentioned above in that the function pi 1 is calculated in a final stage.

[0060] The top 32-bit output data after the function pi2 operation of the operation stage 26 is outputted as the work key K1 32 bits of low rank output data after function pi3 operation is outputted as the work key K2 and the top 32-bit output data after function pi4 operation is outputted as the work key K3. 32 bits of low rank output data after the function pi1 operation of the operation stage 27 is outputted as the work key K4 The top 32-bit output data after function pi2 operation is outputted as work key K5 The top 32-bit output data after function pi3 operation is outputted as the work key K6 the top 32-bit output data after function pi4 operation is outputted as the work key K7 and 32 bits of low rank output data after the function pi1 operation of the final stage 28 is outputted as the work key K8.

[0061] If the key schedule processing which was mentioned above and which is shown in cipher processing shown in drawing 11 and drawing 13 is referred to the composition of four steps of operation stages i.e. an operation algorithm is made equal and it can perform cipher processing or key schedule processing by repeating the operation of four steps of this operation stage and performing it. From this cipher processing or key schedule processing can be performed by repeating an operation core and performing it if an operation core is considered as the composition which carried out cascade connection of the operation stage of the function pi 1 the operation stage of the function pi 2 the operation stage of the function pi 3 and the operation stage of the function pi 4 as shown in drawing 16 (a). This operation core generates a work key



from the data key and system key which are shown in drawing 20 and it is equivalent to Encryptor which is performing cipher processing in the CBC mode and OFB mode and let it be an Encryptor core. In this case the data keys Ks1–Ks4 and the data keys Ks5–Ks8 are supplied to an Encryptor core by time sharing.

[0062] If drawing 12 is referred to the composition of four steps of operation stages, i.e. an operation algorithm is made equal and it can perform decipherment processing by repeating the operation of four steps of this operation stage and performing it. From this decipherment processing can be performed by repeating a decipherment operation core and performing it if a decipherment operation core is considered as the composition which carried out cascade connection of the operation stage of the function pi 4, the operation stage of the function pi 3, the operation stage of the function pi 2, and the operation stage of the function pi 1 as shown in drawing 16 (b). This decipherment operation core is equivalent to Decryptor which is performing decipherment processing in the CBC mode and OFB mode shown in drawing 20 and let it be a Decryptor core. In this case the data keys Ks8–Ks5 and the data keys Ks4–Ks1 are supplied to a Decryptor core by time sharing. Thus by repeating and performing an Encryptor core the algorithm of cipher processing and a key schedule can be performed and a decipherment algorithm can be performed by repeating and performing a Decryptor core.

[0063] As mentioned above although the decipherment device which performs the decoding method of this invention was explained the electronic equipment of this invention is a tuner TV apparatus etc. which are provided with such a decipherment device at least. Although the above explanation explained the cryptogram of 64 bit blocks as what generates the plaintext of 64 bit blocks using a 64 bits data key and a 256-bit system key this invention is not limited to these figures and it can be considered as arbitrary numerical values. This invention is not limited to the encryption and a decoding method which repeats the transposition and substitution which were mentioned above and can be applied to other encryption and decoding methods.

[0064]

[Effect of the Invention] Since this invention is constituted as mentioned above it can be considered as the decoding method that read-out and writing control of a memory means in which the key information for performing decipherment processing is stored can be performed appropriately and memory control can be performed easily. Therefore it can be considered as the decoding method that all the decipherment processings can be performed in real time in a receiver. Since composition of the memory control means in electronic equipment provided with the decoding means which performs such a decoding method can be made small electronic equipment can be provided small and cheaply.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] It is a block diagram showing the example of composition of the decipherment device which is an embodiment of the decoding method of this invention.

[Drawing 2] It is a block diagram showing the composition of IDT in the decipherment device shown in drawing 1.

[Drawing 3] It is a chart showing the memory space of the register in the decipherment device seen from CPU shown in drawing 1.

[Drawing 4] It is a chart showing the logical table of the comparators CP0-CP11 of IDT shown in drawing 2.

[Drawing 5] It is a chart showing the logical table of PE of IDT shown in drawing 2.

[Drawing 6] In the decipherment device shown in drawing 1 it is a figure for explaining how to search a data key with an indirect search method from the information in the header of input data.

[Drawing 7] It is a figure for explaining operation when PID overlaps all over a PID table.

[Drawing 8] It is a figure for explaining operation in case PID applicable all over a PID table does not exist.

[Drawing 9] It is a figure showing the example of the lead timing of DPMEM in the decipherment device shown in drawing 1.

[Drawing 10] It is a flow chart which shows the decipherment flow of the decipherment device shown in drawing 1.

[Drawing 11] It is a figure showing the composition of cipher processing.

[Drawing 12] It is a figure showing the composition of decipherment processing.

[Drawing 13] It is a figure showing the details of the fundamental function in cipher processing.

[Drawing 14] It is a figure showing the detailed composition for calculating the function  $\pi_2$  in a fundamental function.

[Drawing 15] It is a figure showing the composition of key schedule processing.

[Drawing 16] It is a figure showing the composition of an Encryptor core and a Decryptor core.

[Drawing 17] It is a figure showing the algorithm of the conventional encryption.

[Drawing 18] It is a figure showing the algorithm of the conventional decoding.

[Drawing 19] It is a figure showing the composition in the encryption use mode in the CBC mode and OFB mode.

[Drawing 20] It is a figure showing the composition of conventional encryption and decoding method.

[Description of Notations]

1 Received data 2 output data 3 4 switching means 5 Decryptor Six registers and 7 9

and 86 An exclusive "or" circuit and 8 Encryptor10 CPU and 11 An AND gateand 13  
and 14 A comparator and 12 OR gate15 An inverter16 IDT17 DPMEMand 20-28 The  
operation stage39 PE40 ADECand 41-52 A flip-flopand 61-72 A comparatorand 8082  
and 84 [ An adding machineand 81 and 85 ] [ Left round shifter ]

---

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-212090

(43) 公開日 平成9年(1997)8月15日

(51) Int.Cl. <sup>6</sup>	識別記号	片内整理番号	F I	技術表示箇所
G 0 9 C 1/00	6 3 0	7259-5 J	G 0 9 C 1/00	6 3 0 Z
H 0 4 L 9/14			H 0 4 L 9/00	6 4 1

審査請求 未請求 請求項の数 8 F D (全 27 頁)

(21) 出願番号 特願平8-39099

(22) 出願日 平成8年(1996)2月2日

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 江成 正彦

東京都品川区北品川6丁目7番35号 ソニー株式会社内

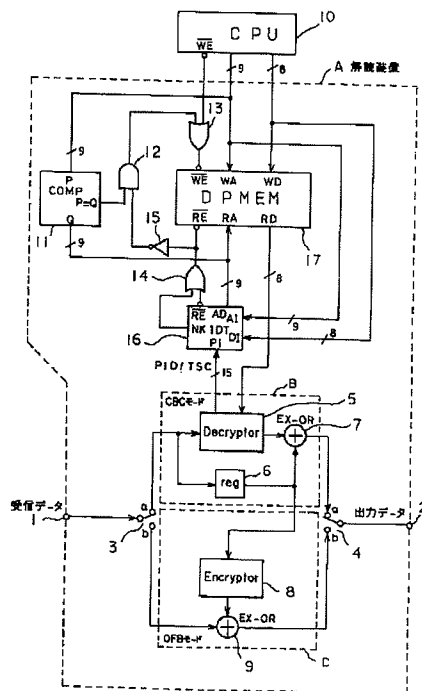
(74) 代理人 弁理士 脇 篤夫 (外1名)

(54) 【発明の名称】 解読化方法および電子機器

(57) 【要約】

【課題】 受信側においてリアルタイムで全ての解読処理を完了することができるようにする。

【解決手段】 端子1に入力された受信データのヘッダ中からPID情報とTSC情報を抽出してIDT16に供給する。IDT16はこれらの情報を受けて間接検索方法によりDPMEM17からデータ鍵をサーチして読み出す。DPMEM17には所定期間毎の更新される複数のデータ鍵が非同期で書き込まれており、DPMEM17の書き込みと読み出しが同じタイミングでかつ同一アドレスについて行われる時に、その書き込みを禁止する。これにより、DPMEM17のメモリ制御を容易に行うことができる。



## 【特許請求の範囲】

【請求項1】 メモリ手段に複数の鍵情報を書き込む書き込みステップと、

入力データ中の情報により指示されたいずれかの鍵情報を、前記メモリ手段から読み出す読み出しステップと、該読み出しステップで読み出された鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記書き込みステップの実行期間と、前記読み出しステップの実行期間とが重なった場合であって、書き込みアドレスと読み出しアドレスとが一致している場合は、前記書き込みステップの実行を禁止するようにしたことを特徴とする解読化方法。

【請求項2】 複数の鍵情報を記憶しているメモリ手段から、入力データ中の指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、該読み出しステップで読み出された鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記読み出しステップでは、前記指示データにより鍵テーブルを参照することにより、前記メモリ手段から読み出す鍵情報をサーチしており、前記指示データに該当する前記鍵情報を読み出すための鍵アドレスが、前記鍵テーブルに2つ以上存在する場合は、最も少ない鍵アドレスを選択することを特徴とする解読化方法。

【請求項3】 複数の鍵情報を記憶しているメモリ手段から、入力データ中の指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、該読み出しステップで読み出された鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記読み出しステップでは、前記指示データによりテーブルを参照することにより、前記メモリ手段から読み出す鍵情報の鍵アドレスをサーチしており、前記指示データにより参照される鍵アドレスが前記テーブルに存在していない場合は、鍵アドレスのサーチを行わないようにしたことを特徴とする解読化方法。

【請求項4】 初期設定を行う初期ステップと、メモリ手段に複数の鍵情報を書き込む書き込みステップと、前記メモリ手段から、入力データ中の指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、該読み出しステップで読み出された鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記読み出しステップでは、前記指示データにより鍵テーブルを参照することにより、前記メモリ手段から読み出す鍵情報の鍵アドレスをサーチしており、初期化期間内においては、サーチした結果、該当する鍵アドレスがないように、前記初期ステップにおいて初期処理されることを特徴とする解読化方法。

【請求項5】 請求項1記載の解読化方法を実行する解読手段を少なくとも備えている電子機器。

【請求項6】 請求項2記載の解読化方法を実行する

解読手段を少なくとも備えている電子機器。

【請求項7】 請求項3記載の解読化方法を実行する解読手段を少なくとも備えている電子機器。

【請求項8】 請求項4記載の解読化方法を実行する解読手段を少なくとも備えている電子機器。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、暗号化されることによりスクランブルされた送信データを受けて、解読するようにした解読化方法および電子機器に関する。

## 【0002】

【従来の技術】通信における情報を秘匿するために、送信情報を暗号化し、暗号化された送信情報を受信して解読することにより、元の情報を得るようにした暗号化・解読化方式が従来から知られている。このような暗号化・解読化方式としては、米国における標準方式であるDES (Data Encryption Standard) 等の暗号アルゴリズムが知られている。

【0003】ところで、暗号アルゴリズムには多種・多様なものがあり、より安全性・高速性に優れた方式が開発されている。この一例として、米国特許第4,982,429号明細書、米国特許第5,103,479号明細書、および特開平1-276189号公報等に記載されている暗号方式(MULTI2方式)が知られている。また、国際標準化機構(ISO)においてもISO9979/0009として登録された暗号化方式や、ISO/IEC10116として登録された暗号化利用モードがある。

【0004】上記MULTI2方式の暗号化方式においては、入力データサイズが64ビット、出力データサイズが64ビットとされており、256ビットサイズのシステム鍵と64ビットのデータ鍵から、暗号化を行うために必要な256ビットサイズのワーク鍵が生成されている。さらに、暗号化段数は正の整数段とされている。このMULTI2方式における暗号化アルゴリズムの概略構成を図17に示す。MULTI2方式は、図17に示すように64ビットのデータ鍵 $K_s$ に256ビットのシステム鍵 $J$ を用いて暗号アルゴリズムの演算を施すことにより256ビットのワーク鍵 $K_w$ を生成する。この暗号アルゴリズムの演算は暗号アルゴリズム実行手段Cにより実行される。生成されたワーク鍵 $K_w$ は、暗号アルゴリズム実行手段Fに供給されて入力された64ビットの平文が暗号化される。なお、暗号アルゴリズム実行手段Cと暗号アルゴリズム実行手段Fとで実行される暗号アルゴリズムは、同一の暗号アルゴリズムである。

【0005】このような暗号化がMULTI2方式の基本的な暗号化アルゴリズムであるが、これでは予め文字、あるいは単語が出現する頻度の分布を統計処理しておき、入手した暗号化文の文字列パターンの頻度分布とのマッチングを取ることにより、平文が推定されてしまうおそれ

がある。そこで、暗号化された64ビットの暗号ブロックと、次に入力される64ビットの入力データとの排他的論理和を演算して暗号文を作成する手法がある。この手法を行って暗号化するモードをCBC (Cipher Block Chaining)モードとよんでいる。前記した暗号アルゴリズム実行手段Fにおいては、このようなCBCモードの暗号アルゴリズムが実行されている。

【0006】また、例えばパケット通信のように通信を行うデータの単位が予め決められている通信方式があるが、64ビットを1ブロックとするようなブロック暗号化方式では、1ブロックのビット数で割り切れないデータ単位が入力された場合に、データが余ってしまうようになる。そこで、その端数処理をOFB (Output Feedback) モードで処理するようにしている。このOFBモードでは、データの端数部分が暗号アルゴリズム実行手段Gに供給され、乱数を使用して暗号化される。この乱数は、ワーク鍵Kwを用いて暗号アルゴリズム実行手段Gにより生成されている。これにより、64ビットを1ブロックとする暗号文を得ることができるようになる。なお、CBCモードおよびOFBモードは暗号化利用モードと呼ばれる。

【0007】また、MULTI2方式における解読化アルゴリズムの概略構成を図18に示す。図18に示すように、64ビットのデータ鍵Ksに256ビットのシステム鍵Jを用いて暗号アルゴリズムの演算を施すことにより256ビットのワーク鍵Kwを生成する。この暗号アルゴリズムの演算は暗号アルゴリズム実行手段cにより実行される。生成されたワーク鍵Kwは、解読アルゴリズム実行手段fに供給されて入力された64ビットの暗号文が解読化される。なお、OFBモードで暗号化されている暗号文は、暗号アルゴリズム実行手段gに供給され、ワーク鍵Kwを用いて暗号アルゴリズム実行手段gにより生成した乱数を使用することにより解読化される。これにより、1ブロック64ビットの暗号文を解読化して64ビットの平文を得ることができる。また、CBCモードとされている場合は、解読アルゴリズム実行手段fがCBCモードの解読アルゴリズムを実行するようにされる。

【0008】ここで、暗号化利用モードの説明を図19を参照しながら行うが、図19(a)にCBCモードの暗号化・解読化の概略構成を示し、図19(b)にOFBモードの暗号化・解読化の概略構成を示している。CBCモードでは、図19(a)に示すようにi番目の平文ブロックM(i)は、排他的論理和回路101に入力され、レジスタ(REG)103により遅延された1ブロック前の暗号文ブロックC(i-1)との排他的論理和が演算される。演算されたデータは暗号アルゴリズム実行手段102において、データ鍵Ksに基づいて生成されたワーク鍵により暗号化される。この暗号化されたi番目の暗号文ブロックC(i)は、

$$C(i) = E_{Ks}(M(i) \oplus C(i-1))$$

と表せる。ただし、 $E_{Ks}(m)$ はmをKsで暗号化することを意味しており、 $\oplus$ は排他的論理和の演算を行うことを示している。

【0009】そして、この暗号文ブロックC(i)は送信され、受信側において受信されることになる。受信された暗号文ブロックC(i)は、解読アルゴリズム実行手段111においてデータ鍵Ksに基づいて生成されたワーク鍵を用いて解読され、排他的論理和回路113に供給される。この排他的論理和回路113にはレジスタ(REG)112において遅延された、1ブロック前の暗号文ブロックC(i-1)が入力されて、両者の排他的論理和が演算される。この時、送信側と受信側のデータ鍵Ksは等しく、これにより、排他的論理和回路113からi番目の平文ブロックM(i)が解読される。i番目の平文ブロックM(i)は次のように表せる。

$$M(i) = D_{Ks}(C(i) \oplus C(i-1))$$

ただし、 $D_{Ks}(c)$ はKsでcを解読化することを示している。

【0010】また、OFBモード時では、i番目の平文ブロックM(i)は排他的論理和回路105に供給される。この排他的論理和回路105には、データ鍵Ksに基づいて生成されたワーク鍵により乱数化された暗号アルゴリズム実行手段104の出力が供給されている。なお、暗号アルゴリズム実行手段104の出力は、レジスタ103により1ブロック遅延されて暗号アルゴリズム実行手段104に戻されている。これにより、排他的論理和回路105からは乱数により暗号化された暗号文ブロックC(i)が出力される。

【0011】そして、この暗号文ブロックC(i)は送信され、受信側において受信されることになる。受信された暗号文ブロックC(i)は、排他的論理和回路114に供給される。この排他的論理和回路114には、暗号アルゴリズム実行手段115においてデータ鍵Ksに基づいて生成されたワーク鍵を用いて乱数化された出力が供給されている。この暗号アルゴリズム実行手段115の出力は、レジスタ(REG)112において1ブロック遅延されて暗号アルゴリズム実行手段115に戻されている。この場合、排他的論理和回路114に供給される乱数は、排他的論理和回路105に供給される乱数と等しく、これにより、排他的論理和回路114からi番目の平文ブロックM(i)が得られる。

【0012】以上説明した暗号化利用モードを有する暗号化・解読化方式の概略構成を図20に示す。この図において、送信側にはスクランブラ100が備えられており、スクランブラ100により入力データがスクランブルされて送信されている。このスクランブルされた送信データは、空間等の伝送路を伝播されて受信側で受信される。受信側には、デスクランブラ110が備えられており、このデスクランブラ110によりスクランブルさ

れた送信データがデスクランブルされて、元のデータに戻され出力されるようになる。

【0013】スクランブラ100は、入力された入力データ（平文）を暗号化する暗号アルゴリズム実行手段であるEncryptor 102と、レジスタ103と、排他的論理和回路（EX-OR）101からなるCBCモード暗号化部と、暗号アルゴリズム実行手段であるEncryptor 104と、排他的論理和回路（EX-OR）105からなるOFBモード暗号化部から構成されている。なお、データ鍵とシステム鍵からワーク鍵を生成するEncryptor 106もスクランブラ100内に備えられている。生成されたワーク鍵はEncryptor 102, 104に供給される。ところで、Encryptor 102、Encryptor 104、Encryptor 106は同一構成とされているので、1つのEncryptorにより3つのEncryptorを兼用することができる。CBCモード暗号化部およびOFBモード暗号化部の動作は前述したとおりであるので、ここでは省略する。

【0014】また、デスクランブラ110は、入力された受信データ（暗号文）を解読化する解読アルゴリズム実行手段であるDecryptor 111と、レジスタ112と、排他的論理和回路（EX-OR）113からなるCBCモード解読化部と、暗号アルゴリズム実行手段であるEncryptor 115と、排他的論理和回路（EX-OR）114からなるOFBモード解読化部から構成されている。なお、データ鍵とシステム鍵からワーク鍵を生成するEncryptor 116もデスクランブラ110内に備えられている。生成されたワーク鍵はDecryptor 111と、Encryptor 115に供給される。なお、Encryptor 115、Encryptor 116は同一構成とされているので、1つのEncryptorにより2つのEncryptorを兼用することができる。また、CBCモード解読化部およびOFBモード解読化部の動作は前述したとおりであるので、ここでは省略する。

【0015】

【発明が解決しようとする課題】ところで、上述したMULTI2方式のような高度な暗号化・解読化方式を、ソフトウェアで実現して使用する場合には、現在の演算手段の演算速度ではリアルタイム処理を行うことができない。すなわち、例えば衛星デジタルテレビジョン放送等に適用した場合は、画像や音声を途切れさせないでリアルタイム再生する必要があることから、受信側における解読処理は高速な処理の可能なハードウェアで行わなければならないことになる。しかしながら、受信側においてリアルタイムで全ての解読処理を完了することができるハードウェアは、複雑な処理を行うことから大型なものになるという問題点があった。

【0016】そこで、本発明は高度な暗号化方式でスクランブルされたデータを、受信側においてリアルタイムで全ての解読処理を小型かつ安価なハードウェアを用い

て完了することができる解読化方法を提供することを第1の目的としている。また、本発明は高度な暗号化方式でスクランブルされたデータを、受信側においてリアルタイムで全ての解読処理を完了することができる解読化手段を備える小型かつ安価な電子機器を提供することを第2の目的としている。

【0017】

【課題を解決するための手段】上記第1の目的を達成するために、本発明の解読化方法は、メモリ手段に複数の鍵情報を書き込む書き込みステップと、入力データ中の情報により指示されたいずれかの鍵情報を、前記メモリ手段から読み出す読み出しステップと、該読み出しステップで読み出された鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記書き込みステップの実行期間と、前記読み出しステップの実行期間とが重なった場合であって、書き込みアドレスと読み出しアドレスとが一致している場合は、前記書き込みステップの実行を禁止するようにしている。

【0018】また、上記第1の目的を達成するために、本発明の他の解読化方法は、複数の鍵情報を記憶しているメモリ手段から、入力データ中の指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、該読み出しステップで読み出された鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記読み出しステップでは、前記指示データにより鍵テーブルを参照することにより、前記メモリ手段から読み出す鍵情報をサーチしており、前記指示データに該当する前記鍵情報を読み出すための鍵アドレスが、前記鍵テーブルに2つ以上存在する場合は、最も少ない鍵アドレスを選択するようにしている。

【0019】さらにまた、上記第1の目的を達成するために、本発明のさらに他の解読化方法は、複数の鍵情報を記憶しているメモリ手段から、入力データ中の指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、該読み出しステップで読み出された鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記読み出しステップでは、前記指示データによりテーブルを参照することにより、前記メモリ手段から読み出す鍵情報の鍵アドレスをサーチしており、前記指示データにより参照される鍵アドレスが前記鍵テーブルに存在していない場合は、鍵アドレスのサーチを行わないようにしている。

【0020】さらにまた、上記第1の目的を達成するために、本発明のさらに他の解読化方法は、初期設定を行う初期ステップと、メモリ手段に複数の鍵情報を書き込む書き込みステップと、前記メモリ手段から、入力データ中の指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、該読み出しステップで読み出された鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記読み出しステップでは、前記指

示データにより鍵テーブルを参照することにより、前記メモリ手段から読み出す鍵情報の鍵アドレスをサーチしており、初期化期間内においては、サーチした結果、該当する鍵アドレスがないように、前記初期ステップにおいて初期処理されるようにしている。

【0021】さらにまた、上記第2の目的を達成する本発明の電子機器は、上記したいずれかの解読化方法を実行する解読手段を少なくとも備えるようにしている。

【0022】このような本発明の解読化方法によれば、解読処理を行うために必要な鍵情報が格納されているメモリ手段の読み出しおよび書き込み制御を適切に行うことができるため、メモリ制御を容易に行うことのできる解読化方法とすることができる。したがって、受信側においてリアルタイムで全ての解読処理を行うことのできる解読化方法とすることができる。また、このような解読化方法を実行する解読手段を備える電子機器においては、メモリ制御手段の構成を簡化することができるので、電子機器を小型かつ安価に提供することができるようになる。

【0023】

【発明の実施の形態】本発明の解読化方法の実施の形態である解読装置の構成例を示すブロック図を図1に示す。この図において、解読装置AはCBCモードの解読化部Bと、OFBモードの解読化部Cと、パケットID(PID)テーブルを備えるデータ鍵間接検索器(IDT)16と、入力データの暗号化に使用されたデータ鍵を解読化部B、Cに渡すデュアルポートメモリ(DPME)17と、DPME17の書き込みアドレスと読み出しアドレスとを比較する比較器(COMP)11から構成されている。

【0024】解読装置Aの端子1には受信データである暗号文データが入力され、この暗号文データは切手段3に入力される。この時、CBCモードとされた場合は、切手段3、および切手段4が共に端子a側に切り換えられ、入力された暗号文データはCBCモード解読化部Bに供給されて解読化されて出力される。また、OFBモードとされた時は、切手段3、および切手段4は共に端子b側に切り換えられて、暗号文データはOFBモード解読化部C側に入力される。そして、入力された暗号文データは、OFBモード解読化部Cにおいて解読化されて平文データが出力される。CBCモード解読化部B、およびOFBモード解読化部Cにおいて、入力された暗号文データは解読処理されて平文データとされるが、その解読化アルゴリズムは、前記図18に示す解読化アルゴリズムと同様である。なお、この場合解読処理に必要なデータ鍵はDPME17から供給される。

【0025】また、端子1に入力される受信データのフォーマットは、例えばISO/IEC13818として規定されているトランスポートストリーム(以下、TS

と記す。)とされる。TSは188バイトのパケット構造とされており、通常4バイトのヘッダのあとに184バイトのペイロードが続くようにされている。さらに、伝送エラーに対するエラー訂正のためにパリティを付加することから、16バイトのパリティ用のダミー期間が付加されて、これらの繰返しストリームとされる。

【0026】このTSのヘッダにはパケットが映像データで構成されているのか、音声データで構成されているのか、あるいは他のデータで構成されているのかを示すPID情報や、暗号化されているTSなのか否かを示すTSC(Transport Scrambling Control)フラグ等が含まれており、図示しない解析部においてヘッダを解析することにより、これらのパケットの属性が解釈されている。この時、TSCにより暗号化されていないパケットと解釈された場合は、解読化処理を施すことなくTSを遅延部を通してそのまま出力するようにする。この遅延部の遅延時間は、解読装置Aが解読処理に要する時間と等しくされる。

【0027】また、1つのPIDに対しては、データ鍵Kse(Ks\_even)とデータ鍵Kso(Ks\_odd)との2つが割り当てられている。これは、データ鍵が数秒ないし数十秒毎に更新されるため、更新に間に合うようにデータ鍵の書き替えを行う必要があること、および、解読処理時にはデータ鍵Kseとデータ鍵Ksoはその一方しか使用されず、使用されていないデータ鍵を更新可能として、そのデータ鍵を更新することができるからである。すなわち、データ鍵を更新する更新制御を容易に行うためである。なお、前記解析部は図示しないが、例えばCBCモード解読化部Bに備えられており、CBCモード解読化部Bから15ビットのデータサイズのPID/TSC情報がIDT16に送られている。

【0028】IDT16においては、受けた13ビットサイズのPID情報を用いてPIDテーブルを参照し、PIDから読出されたアドレス情報と、2ビットサイズのTSC情報とを組み合わせDPME17の読出アドレスRA(データサイズは9ビット)を生成している。生成された読出アドレスRAはAO端子からDPME17のRA端子に供給され、この時にリードイネーブル(RE)信号がアクティブ状態になるよう制御して、読出アドレスRAに該当するデータ鍵をDPME17から読み出している。読み出されたデータ鍵は、入力されたTSの暗号化時に使用されたデータ鍵と等しく、端子RDから出力されてDecryptor5に供給される。Decryptor5は、供給されたデータ鍵と、システム毎に異なるシステム鍵からワーク鍵を生成して、生成されたワーク鍵に基づいて入力された暗号文データの解読処理を行う。なお、システム鍵は予めCPU10から解読装置Aに渡されている。

【0029】また、CPU10は解読化部B、Cで必要とされる更新されるデータ鍵を、解読処理において必要



になる前にDPMEM17に書き込むようにされており、書き込む場合には、ライトイネーブル(WE)をアクティブ状態とすると共に、9ビットのデータサイズの書込アドレス(WA)と、8ビットのデータサイズのデータ鍵の書込データ(WD)をDPMEM17に供給している。この場合、CPU10にはROMやRAMが備えられており、これらのメモリに読取装置Aに与えるデータが書き込まれている。なお、DPMEM17にデータ鍵情報を書き込む時に、書込アドレスと同一の読出アドレスでIDT16がデータ鍵を読み出すようになった場合は、読み出されたデータが不定となることから、この時はDPMEM17への書き込みを禁止している。

【0030】このために、比較器11とアンドゲート12、オアゲート13、14、インバータ15が設けられている。この動作を説明すると、IDT16から出力される読出アドレスRAと、CPU10から出力される書込アドレスWAとが比較器11のP端子およびQ端子にそれぞれ入力される。そして、P端子の入力データとQ端子の入力データとが一致( $P=Q$ )した時に、比較器11からハイレベル信号が出力される。この時、反転REがローレベルでアクティブ状態になっていると、インバータ15からハイレベルが出力されるので、結局のところアンドゲート12からハイレベルが出力されることになる。このアンドゲート12の出力はオアゲート13に入力されて、オアゲート13の出力がハイレベルとされるので、反転WEは非アクティブ状態となり、DPMEM17への書き込みが禁止されることになる。

【0031】次に、CPU10から見た読取装置Aのレジスタのメモリ空間を図3に示すが、9ビットのデータサイズのアドレス(HADD)の上位6ビットが"000100"とされた64bit×1のエリアが、CBCモードの初期値テーブル(CBCInitial value table)に割り当てられている。この初期値テーブルに記憶される初期値は、電源投入時等に実行される初期処理時に、CBCモード読取部Bに供給されてレジスタ6にセットされる。また、アドレス(HADD)の上位4ビットが"0010"とされた256bit×1のエリアが、システム鍵テーブル(SYSTEM\_Key table)に割り当てられている。このシステム鍵はシステム毎に異なるものとされるが、1つのシステムでは固定された鍵である。

【0032】さらに、アドレス(HADD)の上位4ビットが"0100"とされた13bit×12のエリアが、PIDテーブル(PID value table)に割り当てられている。このPIDテーブルはパケット化されたデータ種類を示す情報であり、1チャンネルでは最大12種類とされデータ種類毎に異なるPIDとされる。なお、システム鍵テーブルは所定のレジスタに、PIDテーブルの情報はIDT16に、初期処理時にCPU10から書き込まれる。

【0033】さらにまた、アドレス(HADD)の上位

2ビットが"10"とされた64bit×12のエリアが、データ鍵Kseテーブル(Ks\_even value table)に割り当てられ、アドレス(HADD)の上位2ビットが"11"とされた64bit×12のエリアが、データ鍵Ksoテーブル(Ks\_odd value table)に割り当てられている。このデータ鍵Kseテーブル、およびデータ鍵Ksoテーブルの情報はDPMEM17に書き込まれるが、数秒ないし数10秒の所定タイミング毎に、CPU10により更新されている。

【0034】次に、IDT16の詳細構成の一例を示すブロック図を図2に示す。この図において、41~52はそれぞれ13ビット幅を有する12個のフリップフロップ(DF0~DF11)であり、61~72はそれぞれ13ビット幅を有する12個の比較器(CP0~CP11)である。また、40はアドレスデコーダ(ADEC)であり、9ビットのデータサイズのアドレス情報が入力されると、12本の出力のうちの1本だけに、DF0~DF11のいずれかをアクティブにする出力が出力される。さらに、39はプライオリティ・エンコーダ(PE)であり、TSC情報と組み合わせられるアドレス情報TMP0~TMP3が出力される。

【0035】このIDT16において、端子30にはCPU10から出力された8ビットのデータサイズのPID情報が、13ビットのデータサイズに展開されて入力される。また、端子32にはCPU10から出力された9ビットのデータサイズのアドレス情報が入力され、ADEC40においてデコードされる。このデコード出力はストロブ信号としてDF0~DF11に入力され、DF0~DF11のうちのいずれか1つのみがラッチ可能とされる。この時、入力された13ビットのデータサイズのPID情報は、12個のDF0~DF11に共通に入力されており、ADEC40のデコード出力によりDF0~DF11が順次1つずつ選択されて、供給されているPID情報がラッチされる。このようにして、DF0~DF11に最大12種類のPID情報をラッチすることができ、DF0~DF11によりPIDテーブルが構成されるようになる。

【0036】また、端子33には13ビットのデータサイズのPID情報が入力され、比較器CP0~CP11のB入力端子に共通に入力される。このPID情報は、CBCモード読取部Bにおいて、受信データであるパケットのヘッダから抽出されてIDT16に供給された15ビットのデータサイズのPID/TSC情報のうちの13ビットのデータサイズのPID情報である。この場合、比較器CP0~CP11にはDF0~DF11よりのPID情報がそれぞれA端子に入力されており、端子33より入力されたPID情報と一致するPID情報がA端子に入力されている比較器から一致信号が出力される。従って、PIDテーブルにあるPID情報が端子33から入力されると、CP0~CP11のいずれか1

つから一致信号が出力される。

【0037】このCP0～CP11の論理表を図4に示すが、CP0～CP11のA端子およびB端子にオール“1”が入力されていない時は、一致した時(A=B)に“1”信号が出力される。また、一致してない時は“0”が出力される。さらに、PE39の論理表を図5に示す。このPE39の論理表において、例えば、CP2から“1”信号が出力された時は、PE39からは“0010”の4ビット(TMP0～TMP3)のデータが出力される。このPE39では図5に示すように、D0が最も優先された入力とされ、D11が最下位の優先度の入力とされる。ところで、DF0～DF11に設定されたPIDテーブルのPID情報と一致しないPID情報が入力されることがあるが、この場合にはPEにオール“0”が入力されることになる。この時、PEのNK出力が“1”となる。なお、NK出力が“1”の時には、後述するデータ鍵の鍵テーブルを読みに行くことが禁止される。

【0038】また、電源投入時等の初期処理時には端子31のレベルが“0”となり、DF0～DF11が全てプリセットされることから、DF0～DF11から“1”が出力される。従って、この場合はCP0～CP11からオール“0”が出力されることになり、上述のようにデータ鍵の鍵テーブルを読みに行くことが禁止される。なお、鍵テーブルを読みに行くことが禁止されている時は、鍵テーブルが使用されないで、この鍵テーブルにCPU10からデータ鍵を書き込むことができる。すなわち、データ鍵の初期設定をすることができる。また、端子31はパワーオンリセット端子であり、所定時間後にそのレベルは“1”に復帰する。

【0039】次に、受信データ中のPID情報とTSC情報により、データ鍵の鍵テーブルをサーチする動作を図6を参照しながら説明する。図6に示すステップS20にて、受信データのヘッダから13ビットのデータサイズのPID情報として“PID-F”が、2ビットのデータサイズのTSC情報として“10”が入力されたとする。この“10”は、データ鍵Ksでスクランブルされていることを示しているものとする。次いで、ステップS21にてこのPID情報はCP0～CP11のA端子に入力されて、DF0～DF11に格納されているPID情報と比較される。この結果、CP5から“1”信号が出力され、PE39のTEMP3～TEMP0の出力TMP[3..0]が、“0101”となる。すなわち、“PID-F”がサーチされる。この、PE39の出力である“0101”は、アドレスHADDの第4ビットから第1ビットとされる。

【0040】次いで、ステップS22にてTSC情報TSC[1..0]の2ビット(“10”)がHADDの第5ビット、第6ビットとされて、6ビットサイズのアドレスHADD[8..3]が生成される。したがっ

て、アドレスHADD[8..3]は“100101”となる。そして、ステップS23にて、生成された“100101”のアドレスHADD[8..3]により鍵テーブルを参照すると、Ks\_even Table から64ビット幅のデータ鍵Ks-e-Fが得られるようになる。そして、得られたデータ鍵Ks-e-Fに基づいて入力された暗号文の解読が解読装置Aで実行される。このように、本発明の解読化方法では間接的にデータ鍵を検索する間接検索方法としているので、PID情報に対応する13ビット幅のテーブルを用意することなく、6ビット幅の鍵テーブルを用意すればよいのでメモリの容量を削減して小型化することができる。

【0041】なお、このように解読中では鍵テーブルのKs\_even Table, Ks\_odd Table の一方しか使用されないため、使用していない鍵テーブルのデータ鍵を更新することができる。これは、前述したようにCPU10が実行するが、CPU10のタイミングと解読装置Aのタイミングとは非同期で動作するので、CPU10は非同期で更新するデータ鍵を書き込みに行く。このため、鍵テーブルの格納されるデュアルポートメモリ17においては、前述したように書き込みと読み出しが同時に同一アドレスで行われる場合が生じるのである。この時のタイミング例を図9に示すが、図示するタイミングで8クロック幅の反転REが発生した場合には、この8クロック期間においては、前述したようにCPU10からの同一アドレスの書き込みが禁止される。

【0042】ところで、何らかの原因によりPIDテーブルに同一のPIDが格納されている場合が生じる。例えば、1チャンネル当り最大12種類のPID情報とされるが、12種類のPIDを必要としない場合はPIDテーブルには12種類のPIDを書き込む必要はなく、必要する種類のPIDだけを書き込むことになる。すると、書き込まれていないPIDの欄のデータが偶然PIDと同じデータになることがある。このような場合には、誤ったデータ鍵を読み出して解読できなくなってしまう恐れがあるので、本発明においては次のようにしてこれを防止している。

【0043】図7に示すようにPIDテーブルに複数の“PID-F”がある場合は、その内のアドレスHADDが小さい方を優先するようにしている。これは、アドレスHADDの小さい方からPIDテーブルに書き込みに行くので、アドレスHADDの大きい方が誤っている確率が高いからである。これにより、PID情報として“PID-F”が入力されると、アドレスHADD[8..1]として“01000101”が得られるようになり、前述した処理と同様の処理が行われ、Ks\_even Table から64ビット幅のデータ鍵Ks-e-Fが得られるようになる。

【0044】また、入力データのヘッダからのPID情報により、PIDテーブルを参照しても該当するPID

がない場合がある。例えば、図8に示すようにPID情報として“PID-F”が入力されても、PIDテーブル中には“PID-F”に該当するPIDがない。このような場合には、鍵テーブルのKs\_even Table, Ks\_odd Table のいずれも読みに行くことなく、データ鍵を読み出さない。この場合は、入力データであるTSはそのまま出力することになる。

【0045】次に、図1に示す解読装置Aの解読フローを図10に示すが、トランスポートストリーム(TS)が入力されると、ステップS10にてスクランブルされているかが判定される。この判定はスクランブルされたことを示すヘッダ中のフラグが立っているか否かを検出することにより判定される。この場合、フラグが立っている場合はスクランブルonと判定されて、ステップS11に進み、ここで所望のフラグ等の書き換えが行われる。次いで、ステップS12にてヘッダからPID情報が抽出されて、鍵テーブルが参照される。この場合の鍵テーブルは、鍵テーブル処理のステップS16にてホストインターフェース処理が行われて、ステップS17にて書き込まれた鍵テーブルが参照される。なお、ステップS12の処理は、前記図6に示す処理である。

【0046】以上のステップS10ないしステップS12の処理がヘッダコントロール処理である。なお、ステップS10にてスクランブルoffと判定された場合には、TSはそのまま出力される。次いで、ステップS13にてCBCモードの解読処理が行われ、ステップS14にて解読処理される暗号文が64ビットの整数倍か否かが判定される。暗号文に端数がありNoと判定された場合は、ステップS15にて端数部分についてOFBモードの解読処理が行われ、解読処理された平文が出力される。また、ステップS14にて64ビットの整数倍と判定された場合は、解読処理された64ビットの平文が出力されるようになる。

【0047】ところで、図10に示すような解読フローの実行は、前記図18に示す解読化アルゴリズムにより実行されている。図18に示す解読化アルゴリズムは前述したとおりであるのでここでは省略するが、解読化アルゴリズム中の解読アルゴリズムおよび暗号アルゴリズムを実行する構成の詳細を図11ないし図15を参照して説明する。図11は暗号アルゴリズムを実行する暗号処理の構成を示す。図11において、64ビット幅の入力データは、上位32ビットのデータと下位の32ビットのデータに分割されて最初の暗号8段に入力される。この暗号8段は、関数の演算を行う4段の演算段が2回繰り返された構成とされる。そして、入力された上位32ビットのデータと下位の32ビットのデータに、演算段20の初段において関数 $\pi$ 1の演算が施される。ついで、第2段において初段の出力に関数 $\pi$ 2の演算が施される。この場合、第2段には32ビット幅のワーク鍵K1が入力され、このワーク鍵K1を用いて第2段の演算

が行われている。

【0048】さらに、第3段において第2段の出力に関数 $\pi$ 3の演算が施される。この場合、第3段には32ビット幅のワーク鍵K2, K3が入力され、このワーク鍵K2, K3を用いて演算が行われている。続いて、第4段において第3段の出力に関数 $\pi$ 4の演算が施される。この場合、第4段には32ビット幅のワーク鍵K4が入力され、このワーク鍵K4を用いて演算が行われている。さらに残る4段の演算を行う演算段21において、演算段20からの出力に初段において関数 $\pi$ 1の演算が施される。ついで、第2段において初段の出力に関数 $\pi$ 2の演算が施される。この場合、第2段には32ビット幅のワーク鍵K5が入力され、このワーク鍵K5を用いて演算が行われている。

【0049】さらに、第3段において第2段の出力に関数 $\pi$ 3の演算が施される。この場合、第3段には32ビット幅のワーク鍵K6, K7が入力され、このワーク鍵K6, K7を用いて演算が行われている。続いて、第4段において第3段の出力に関数 $\pi$ 4の演算が施される。この場合、第4段には32ビット幅のワーク鍵K8が入力され、このワーク鍵K8を用いて演算が行われている。このようにして暗号処理の行われた上位32ビット、下位32ビットの合計64ビット幅のデータは、さらに暗号8段22に入力される。この暗号8段22において、上述した暗号8段の演算と同様の演算が施されて、上位32ビット、下位32ビットの合計64ビット幅のランダム化された出力データが得られる。

【0050】なお図示しているように、暗号8段の繰返し数は2回に限らず、所望の回数繰り返すことができる。この回数を多く繰り返すほど、出力データは高度にランダム化されて、暗号強度を強いものとしてできる。なお、演算段の格段で行われている関数の演算は、一定の規則に従ってある文字を他の文字に置き換える換字と、文字の順序を入れ替える転置を行う演算とされている。

【0051】次に、解読アルゴリズムを実行する解読処理の構成を図12に示すが、前述した暗号処理と異なる点は、暗号処理の構成の出力側から逆に演算を行うようにしている点である。すなわち、暗号8段のうちの最初の4段の演算段23においては、上位32ビットと下位32ビットに分割された64ビット幅の暗号化されている入力データに、32ビット幅のワーク鍵K8を用いて関数 $\pi$ 4の演算を施している。次いで、第2段目において、初段の出力データにワーク鍵K7を用いて関数 $\pi$ 3の演算を施している。さらに、第3段目において、第2段の出力データにワーク鍵K6, K7を用いて関数 $\pi$ 2の演算を施している。さらにまた第3段目において、第2段の出力データにワーク鍵K5を用いて関数 $\pi$ 2の演算を施し、第4段目において、第3段の出力データに関数 $\pi$ 1の演算を施している。

【0052】このような4段の演算が、演算24においてワーク鍵K4～K1を用いて同様に行われる。さらに、上記した暗号8段の演算が継続されている暗号8段25においても実行されて、解読化された上位32ビット、下位32ビットの計64ビット幅の出力データが得られるようになる。なお、暗号8段の繰返し回数は、暗号処理において実行された暗号8段の繰返し回数と同じ回数とされる。

【0053】次に、演算段で行われている演算の詳細を暗号処理の演算段20を例に上げて図13を参照しながら詳細に説明する。初段の関数 $\pi 1$ の演算では、入力された32ビットに分割された上位ビットは、演算されることなくそのまま出力され、上位ビットと下位ビットの排他的論理和が演算されて下位ビットとして出力される。続く、第2段の関数 $\pi 2$ の演算では、下位32ビットのデータxにワーク鍵K1が加算されて、 $x + K1$ がまず演算される。次いで、 $x + K1$ をyとした時に、yを1ビット左巡回シフトし、その値にy-1を加算してzを得る。次に、zを4ビット左巡回シフトし、その値とzとの排他的論理和を得る。この演算結果と、上位32ビットの排他的論理和が演算されて、演算された上位32ビットのデータが出力される。この場合、下位32ビットは入力されたデータが、演算されることなくそのまま出力される。

【0054】また、第3段の関数 $\pi 3$ の演算では、上位32ビットのデータxにワーク鍵K2が加算されて、 $x + K2$ がまず演算される。次いで、 $x + K2$ をyとした時に、yを2ビット左巡回シフトし、その値にy+1を加算してzを得る。次に、zを8ビット左巡回シフトし、その値とzとの排他的論理和aを得る。さらに、aにワーク鍵K3が加算されて、 $a + K3$ が演算される。次いで、 $a + K3$ をbとした時に、bを1ビット左巡回シフトし、その値に-bを加算してcを得る。次に、aとxのビット毎の論理和と、cを16ビット左巡回シフトした値との排他的論理和を演算する。この演算結果と、下位32ビットのデータとの排他的論理和を演算して、演算された下位32ビットのデータを出力する。なお、上位32ビットのデータは、演算されることなくそのまま上位32ビットの出力データとなる。

【0055】さらにまた、第4段の関数 $\pi 4$ の演算では、下位32ビットのデータxにワーク鍵K4が加算されて、 $x + K4$ がまず演算される。次いで、 $x + K4$ をyとした時に、yを2ビット左巡回シフトし、その値にy+1を加算する。この演算結果と、上位32ビットの排他的論理和が演算されて、演算された上位32ビットのデータが出力される。この場合、下位32ビットのデータは演算されることなく、そのまま下位32ビットのデータとして出力される。

【0056】上記演算において、ワーク鍵K1～K4をデータに加算することにより、文字を他の文字で置き換

える換字処理が行われ、データを巡回シフトさせることにより文字の位置を入れ替える転置が行われる。このように、換字と転置のアルゴリズムを行うことにより平文が暗号文に暗号化される。また、解読化する場合には、暗号化と逆の換字と転置のアルゴリズムを行うことにより元の平文に解読することができる。

【0057】次に、上述した関数の演算を行う構成をさらに詳細に説明するが、関数 $\pi 2$ の例を図14に上げて説明するものとする。図14において、第1加算器Add80において、下位32ビットの入力データxと32ビットのワーク鍵K1とが加算され、加算データyが出力される。この加算データyは第1左巡回シフター81において1ビット左巡回シフトされると共に、第2加算器82において、第1左巡回シフター81の出力と加算される。この加算結果に第3加算器84において-1が加算されて、データzが演算される。このデータzは第2左巡回シフター85において4ビット左巡回シフトされると共に、排他的論理和回路86に供給される。この排他的論理和回路86には第2左巡回シフター85の出力データ、データz、上位32ビット入力データが入力され、3つのデータの排他的論理和が演算される。この演算結果は、次段に入力される上位32ビット入力データとなる。また、下位32ビット入力データは、演算されることなく次段に入力される下位32ビット入力データとなる。

【0058】次に、64ビット幅のデータ鍵と256ビット幅のシステム鍵から256ビット幅のワーク鍵を生成する鍵スケジュール処理の構成を図15に示す。鍵スケジュール処理は図15に示すように4段の演算段26、27が2段と、1段の演算段28が1段縦続接続された構成とされている。また、4段の演算段26、27においては、初段において関数 $\pi 1$ の演算が行われ、2段目において関数 $\pi 2$ の演算が行われ、3段目において関数 $\pi 3$ の演算が行われ、4段目において関数 $\pi 4$ の演算が行われている。

【0059】このような演算アルゴリズムは、前述した暗号処理のアルゴリズムと同じであるのでその説明は省略するが、鍵スケジュール処理においては、入力データが64ビットのデータ鍵とされ、それぞれ32ビットのシステム鍵J1～J8を用いて関数 $\pi 1$ ないし関数 $\pi 4$ の演算が行われて、それぞれ32ビットの8つのワーク鍵K1～K8が生成されている。ただし、全体で9段の演算を行っており、最終段において関数 $\pi 1$ の演算を行う点で、前述した暗号処理のアルゴリズムと相違している。

【0060】なお、演算段26の関数 $\pi 2$ 演算後の上位32ビット出力データがワーク鍵K1として出力され、関数 $\pi 3$ 演算後の下位32ビット出力データがワーク鍵K2として出力され、関数 $\pi 4$ 演算後の上位32ビット出力データがワーク鍵K3として出力されている。さら

に、演算段 27 の関数  $\pi 1$  演算後の下位 32 ビット出力データがワーク鍵 K 4 として出力され、関数  $\pi 2$  演算後の上位 32 ビット出力データがワーク鍵 K 5 として出力され、関数  $\pi 3$  演算後の上位 32 ビット出力データがワーク鍵 K 6 として出力され、関数  $\pi 4$  演算後の上位 32 ビット出力データがワーク鍵 K 7 として出力され、最終段 28 の関数  $\pi 1$  演算後の下位 32 ビット出力データがワーク鍵 K 8 として出力されている。

【0061】 上述した、図 11 に示す暗号処理と図 13 に示す鍵スケジュール処理を参照すると、4 段の演算段の構成、すなわち演算アルゴリズムは等しくされており、この 4 段の演算段の演算を繰り返し行うことにより、暗号処理あるいは鍵スケジュール処理を実行することができる。このことから、演算コアを図 16 (a) に示すように、関数  $\pi 1$  の演算段、関数  $\pi 2$  の演算段、関数  $\pi 3$  の演算段、関数  $\pi 4$  の演算段を縦続接続した構成とすれば、演算コアを繰返し実行することで、暗号処理あるいは鍵スケジュール処理を実行することができる。なお、この演算コアは、図 20 に示すデータ鍵とシステム鍵からワーク鍵を生成すると共に、CBC モードおよび OFB モードで暗号処理を行っている Encryptor に相当し、Encryptor コアとされる。この場合、Encryptor コアにはデータ鍵 K s 1 ~ K s 4 とデータ鍵 K s 5 ~ K s 8 とが時分割で供給される。

【0062】 また、図 12 を参照すると、4 段の演算段の構成、すなわち演算アルゴリズムは等しくされており、この 4 段の演算段の演算を繰り返し行うことにより、解読処理を実行することができる。このことから、解読演算コアを図 16 (b) に示すように、関数  $\pi 4$  の演算段、関数  $\pi 3$  の演算段、関数  $\pi 2$  の演算段、関数  $\pi 1$  の演算段を縦続接続した構成とすれば、解読演算コアを繰返し実行することで、解読処理を実行することができる。なお、この解読演算コアは、図 20 に示す CBC モードおよび OFB モードの解読処理を行っている Decryptor に相当し、Decryptor コアとされる。この場合、Decryptor コアにはデータ鍵 K s 8 ~ K s 5 とデータ鍵 K s 4 ~ K s 1 とが時分割で供給される。このように、Encryptor コアを繰返し実行することにより暗号処理および鍵スケジュールのアルゴリズムを実行することができ、Decryptor コアを繰返し実行することにより解読アルゴリズムを実行することができる。

【0063】 以上、本発明の解読化方法を実行する解読装置の説明をしたが、本発明の電子機器は、このような解読装置を少なくとも備えているチューナーやテレビジョン装置等である。また、以上の説明では 64 ビットブロックの暗号文を 64 ビットのデータ鍵、および 256 ビットのシステム鍵を用いて 64 ビットブロックの平文を生成するものとして説明したが、本発明がこれらの数値に限定されるものではなく、任意の数値とすることができる。さらに、本発明は上述した転置および換字を繰

り返すような暗号化・解読化方式に限定されるものではなく、他の暗号化・解読化方式にも適用することができる。

#### 【0064】

【発明の効果】 本発明は以上のように構成されているので、解読処理を行うための鍵情報が格納されているメモリ手段の読み出しおよび書き込み制御を適切に行うことができ、メモリ制御を容易に行うことのできる解読化方法とすることができる。したがって、受信側においてリアルタイムで全ての解読処理を実行することのできる解読化方法とすることができる。また、このような解読化方法を実行する解読手段を備える電子機器におけるメモリ制御手段の構成を小さくすることができるので、電子機器を小型かつ安価に提供することができるようになる。

#### 【図面の簡単な説明】

【図 1】 本発明の解読化方法の実施の形態である解読装置の構成例を示すブロック図である。

【図 2】 図 1 に示す解読装置における IDT の構成を示すブロック図である。

【図 3】 図 1 に示す CPU から見た解読装置におけるレジスタのメモリ空間を示す図表である。

【図 4】 図 2 に示す IDT の比較器 CP0 ~ CP11 の論理表を示す図表である。

【図 5】 図 2 に示す IDT の PE の論理表を示す図表である。

【図 6】 図 1 に示す解読装置において、入力データのヘッダ中の情報から間接検索方法によりデータ鍵をサーチする方法を説明するための図である。

【図 7】 PID テーブル中に PID が重複している場合の動作を説明するための図である。

【図 8】 PID テーブル中に該当する PID が存在しない場合の動作を説明するための図である。

【図 9】 図 1 に示す解読装置における DP MEM のリードタイミングの例を示す図である。

【図 10】 図 1 に示す解読装置の解読フローを示すフローチャートである。

【図 11】 暗号処理の構成を示す図である。

【図 12】 解読処理の構成を示す図である。

【図 13】 暗号処理における基本関数の詳細を示す図である。

【図 14】 基本関数中の関数  $\pi 2$  を演算するための詳細な構成を示す図である。

【図 15】 鍵スケジュール処理の構成を示す図である。

【図 16】 Encryptor コアと Decryptor コアの構成を示す図である。

【図 17】 従来の暗号化のアルゴリズムを示す図である。

【図 18】 従来の解読化のアルゴリズムを示す図である。

【図19】CBCモードとOFBモードの暗号化利用モードの構成を示す図である。

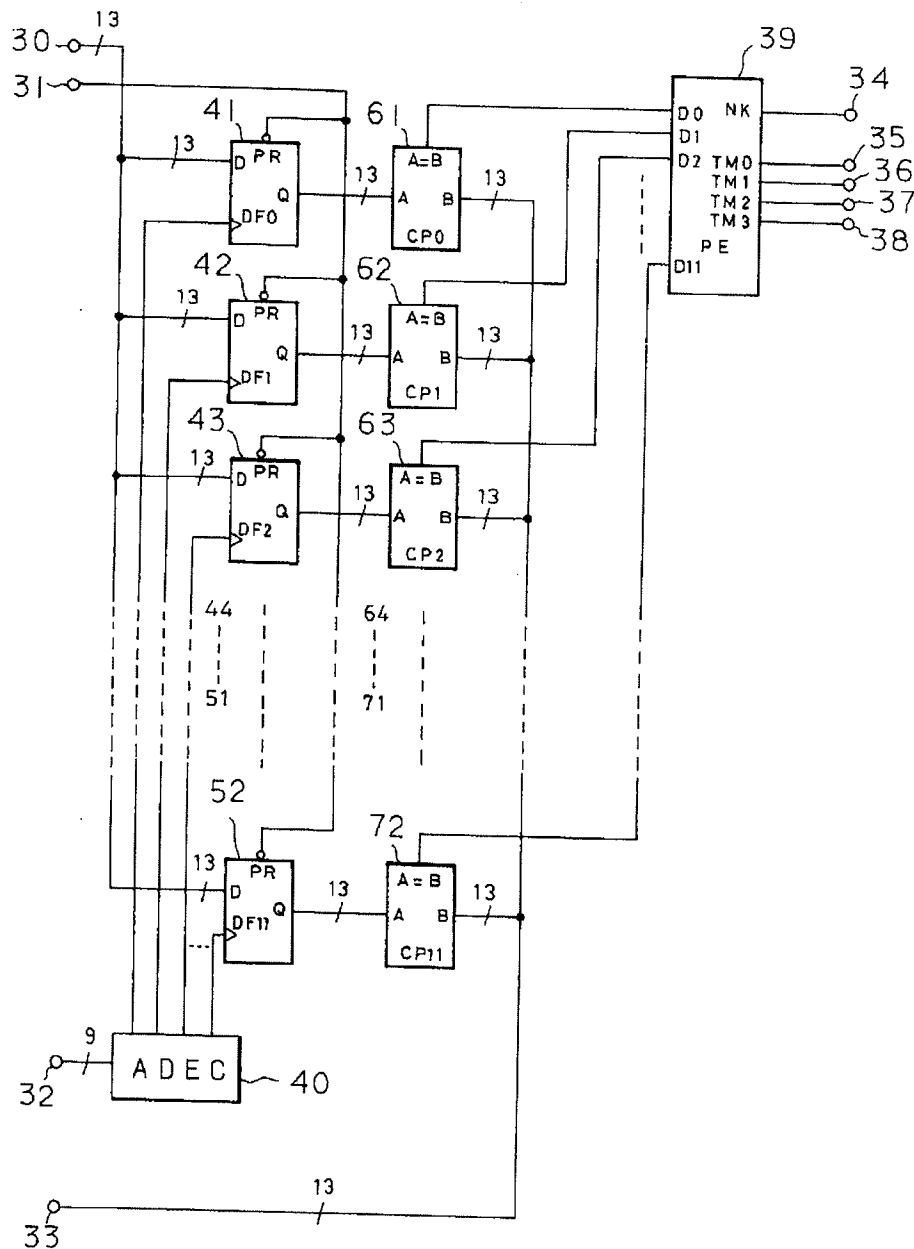
【図20】従来の暗号化・解読化方式の構成を示す図である。

【符号の説明】

1 受信データ、2 出力データ、3、4 切り換え手段、5 Decryptor、6レジスタ、7、9、8 6 排他

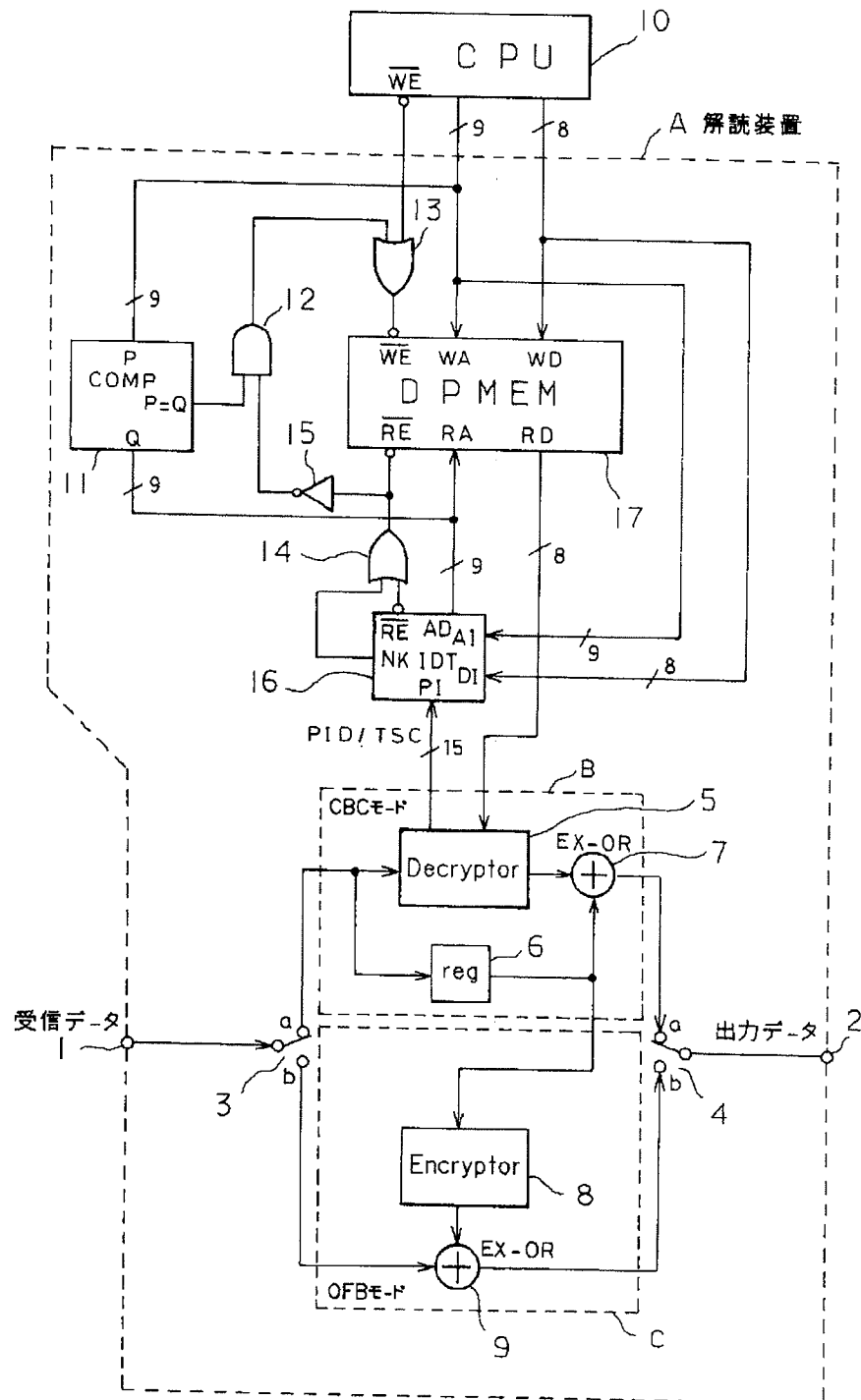
的論理和回路、8 Encryptor、10 CPU、11 比較器、12 アンドゲート、13、14 オアゲート、15 インバータ、16 IDT、17 DPME M、20~28 演算段、39 PE、40 ADE C、41~52 フリップフロップ、61~72 比較器、80、82、84 加算器、81、85 左巡回シフター

【図2】



IDTブロック図

【図1】



【図3】

HADD[8..0]	内容	ビット数
0 0010 0xxx	CBC Initial value table	64bit×1
0 010x xxxx	SYSTEM_Key table	256bit×1
0 100x xxxx	PID value table	13bit×12
1 0xxx xxxx	Ks_even value table	64bit×12
1 1xxx xxxx	Ks_odd value table	64bit×12

【図4】

CP0～CP11の論理表

入力 A, B	出力 A=B
A=B (AもBもall '1'でないとき)	1
A=B (AもBもall '1'のとき)	0
A<B	0
A>B	0

【図5】

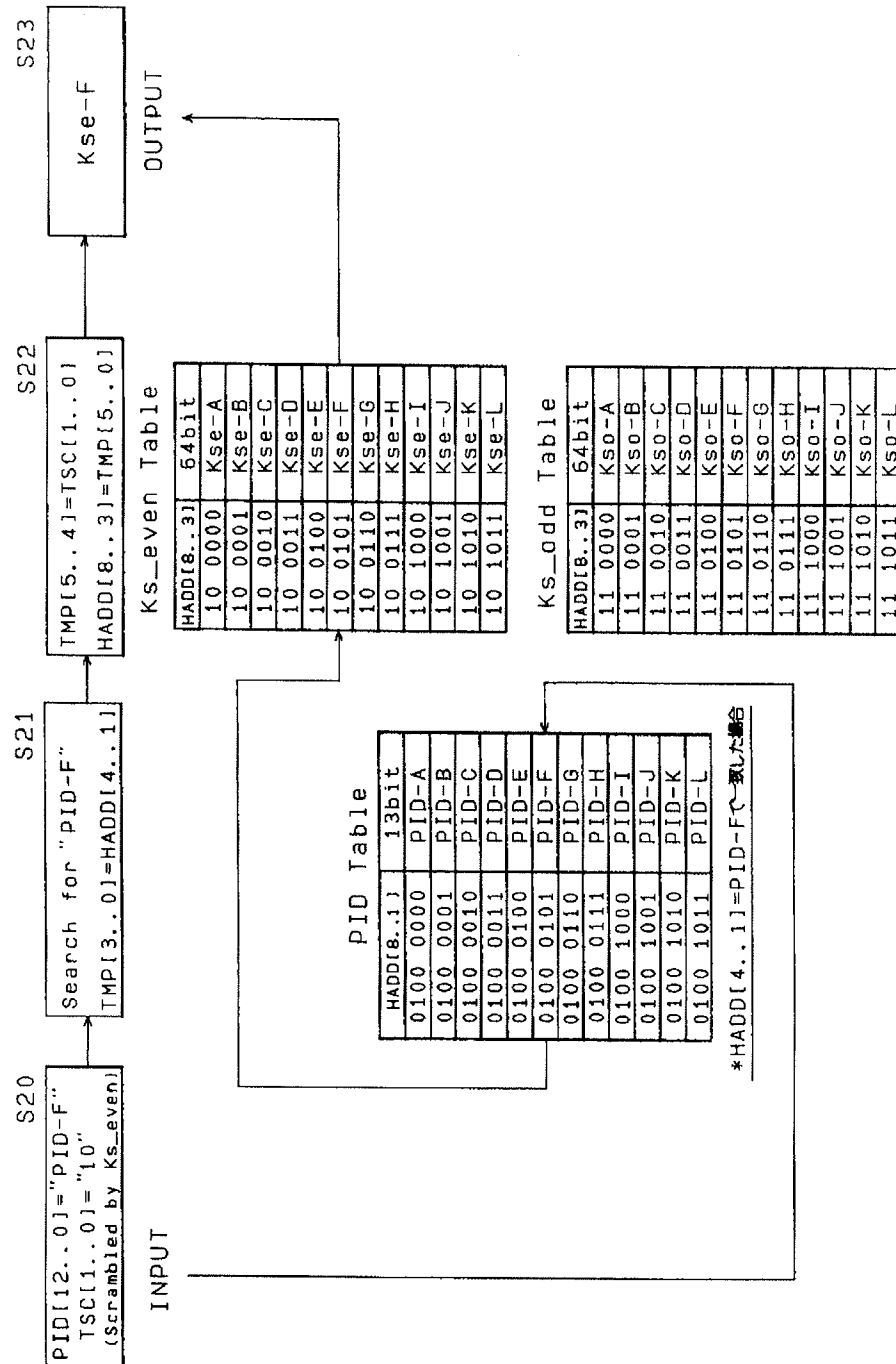
PEの論理表

入 力												出 力				
D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	TMP3	TMP2	TMP1	TMP0	NK
x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0
x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	1	0
x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	1	0
x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	1	0
x	x	x	x	x	x	x	1	0	0	0	0	0	0	1	0	0
x	x	x	x	x	x	1	0	0	0	0	0	0	0	1	0	1
x	x	x	x	x	1	0	0	0	0	0	0	0	0	1	0	0
x	x	x	x	1	0	0	0	0	0	0	0	0	0	1	1	0
x	x	x	1	0	0	0	0	0	0	0	0	0	0	1	0	0
x	x	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
x	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

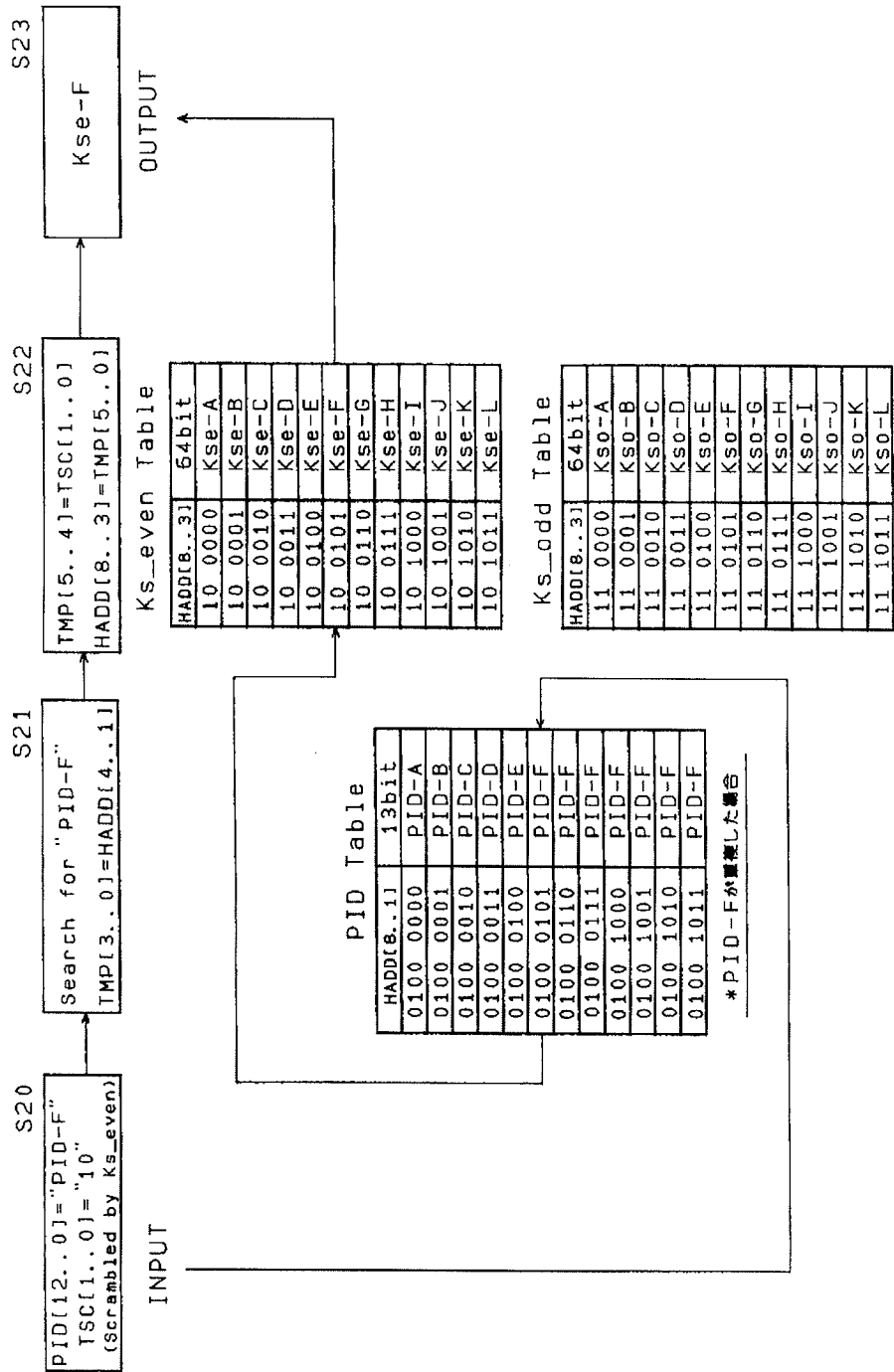
x=Do not care



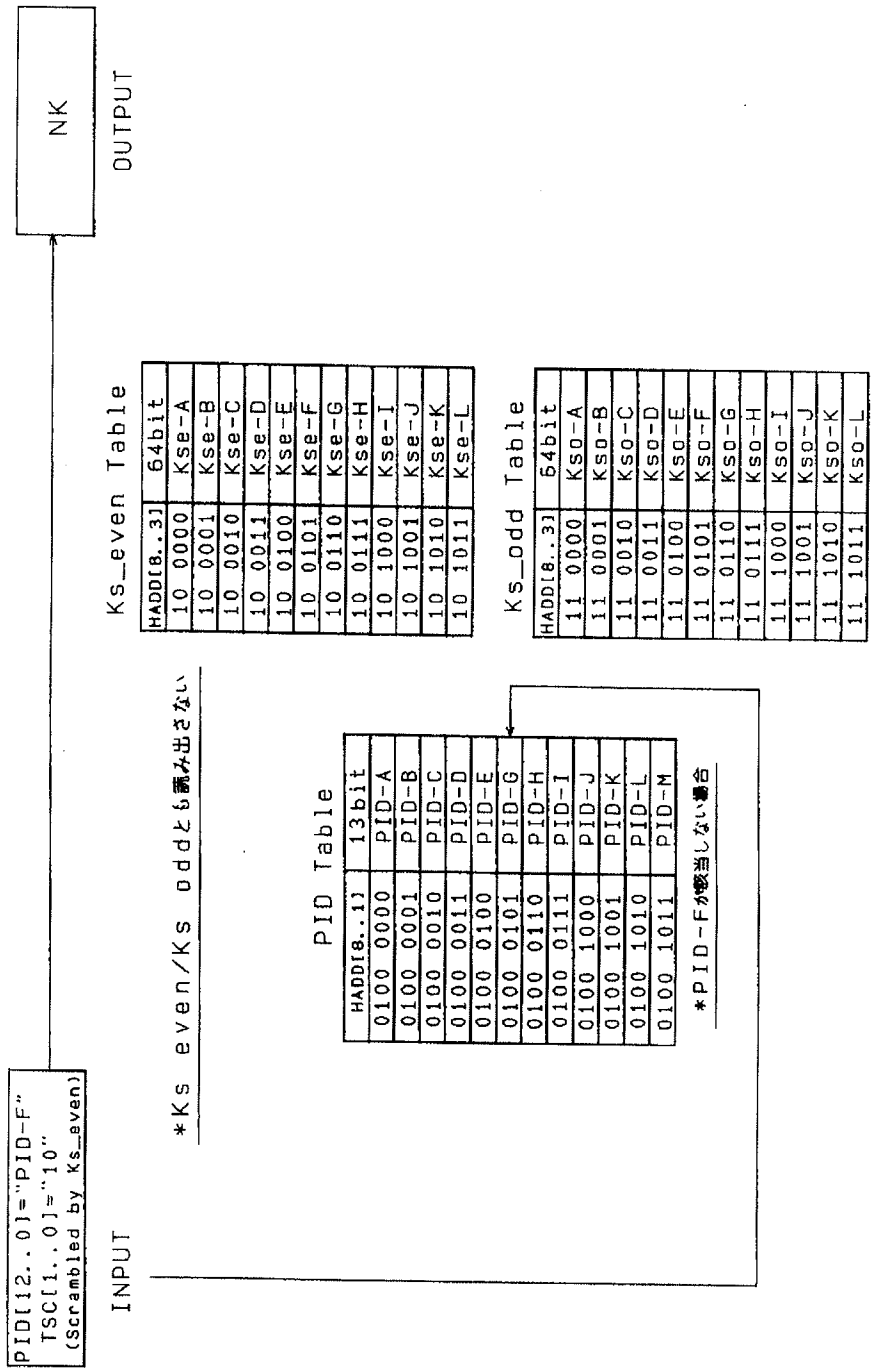
【図6】



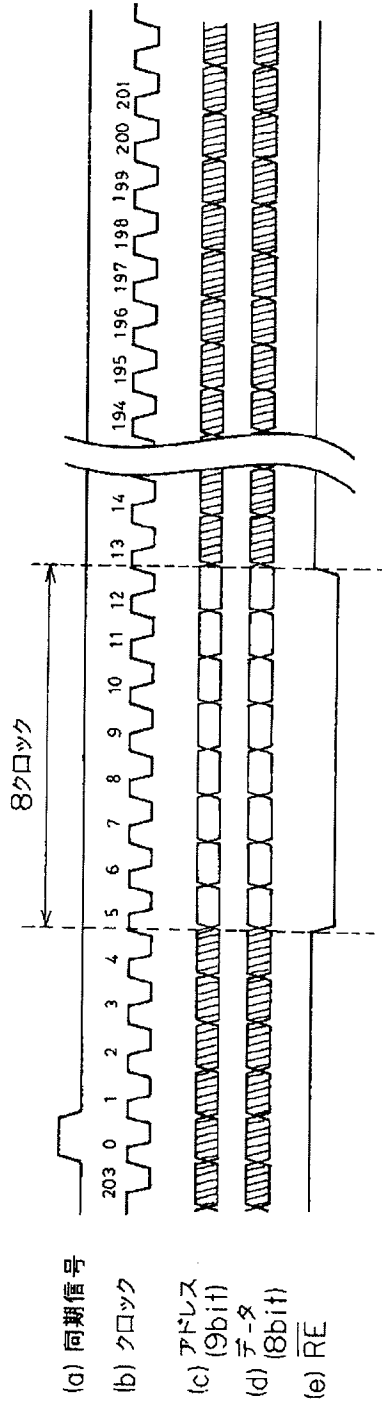
【図7】



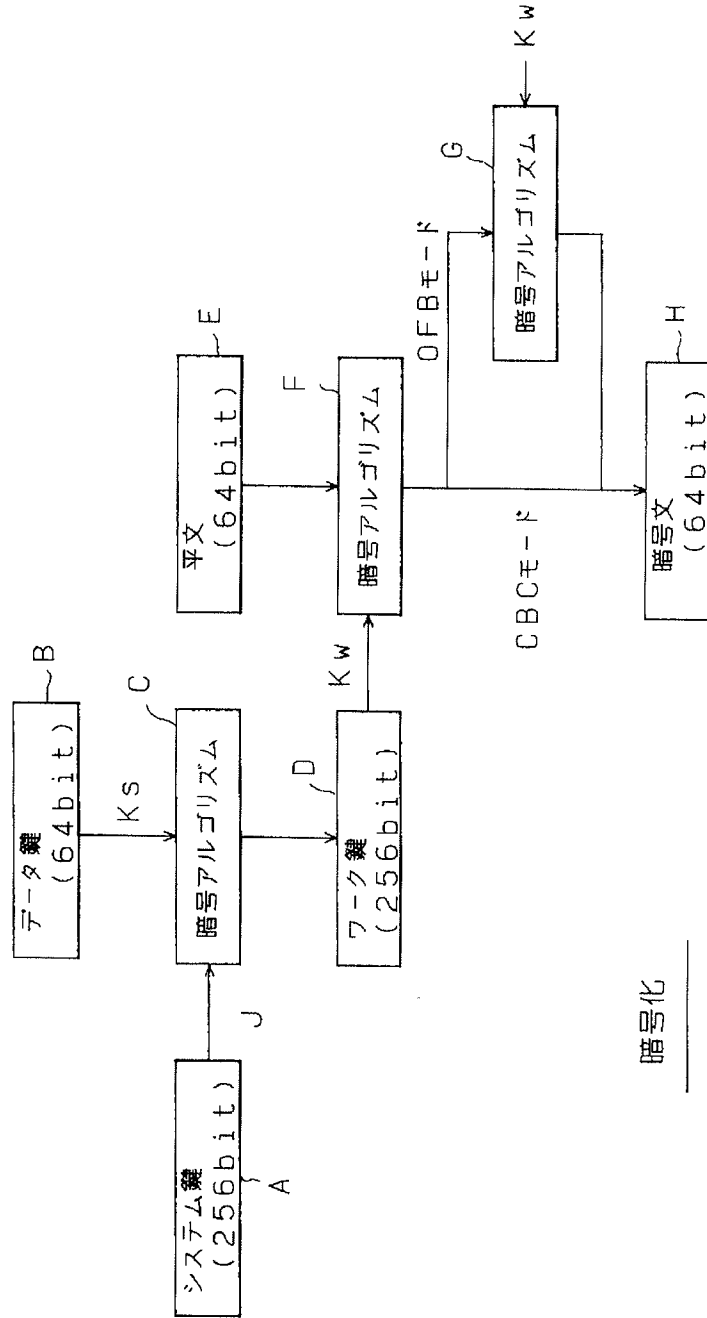
【図8】



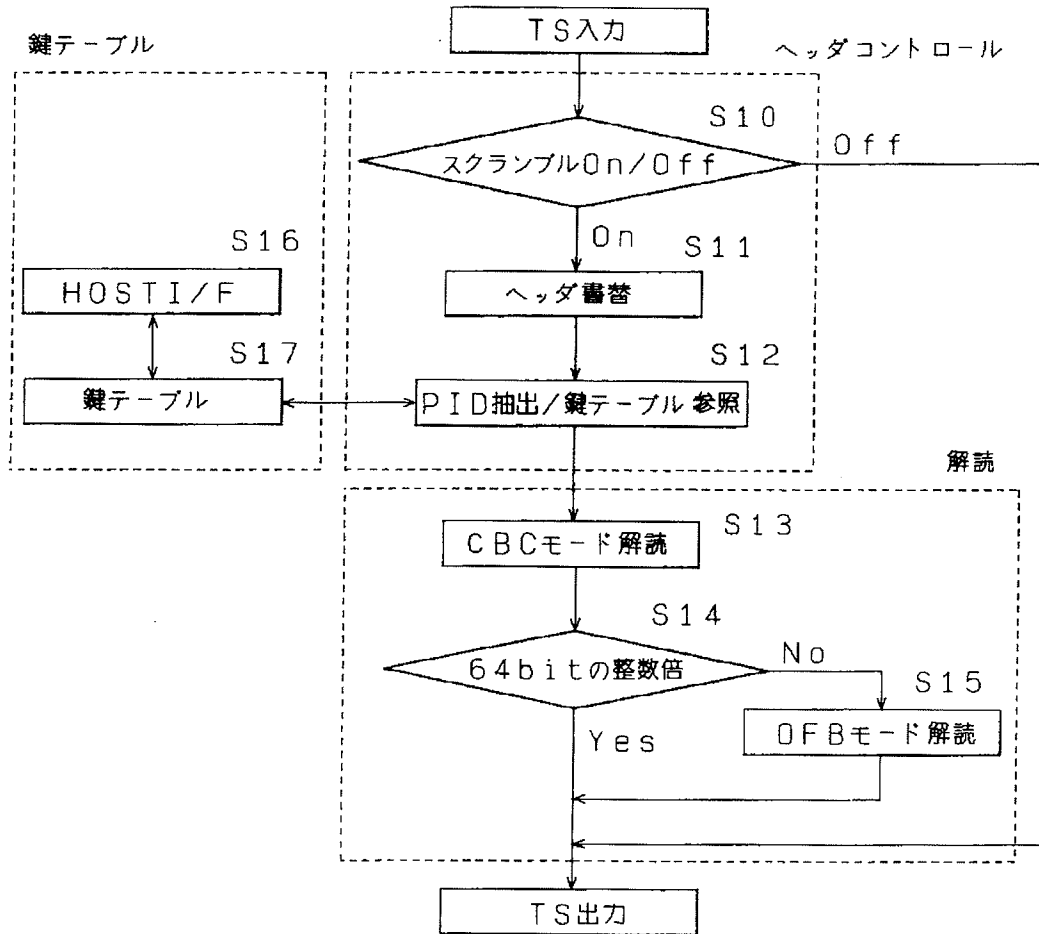
【図9】



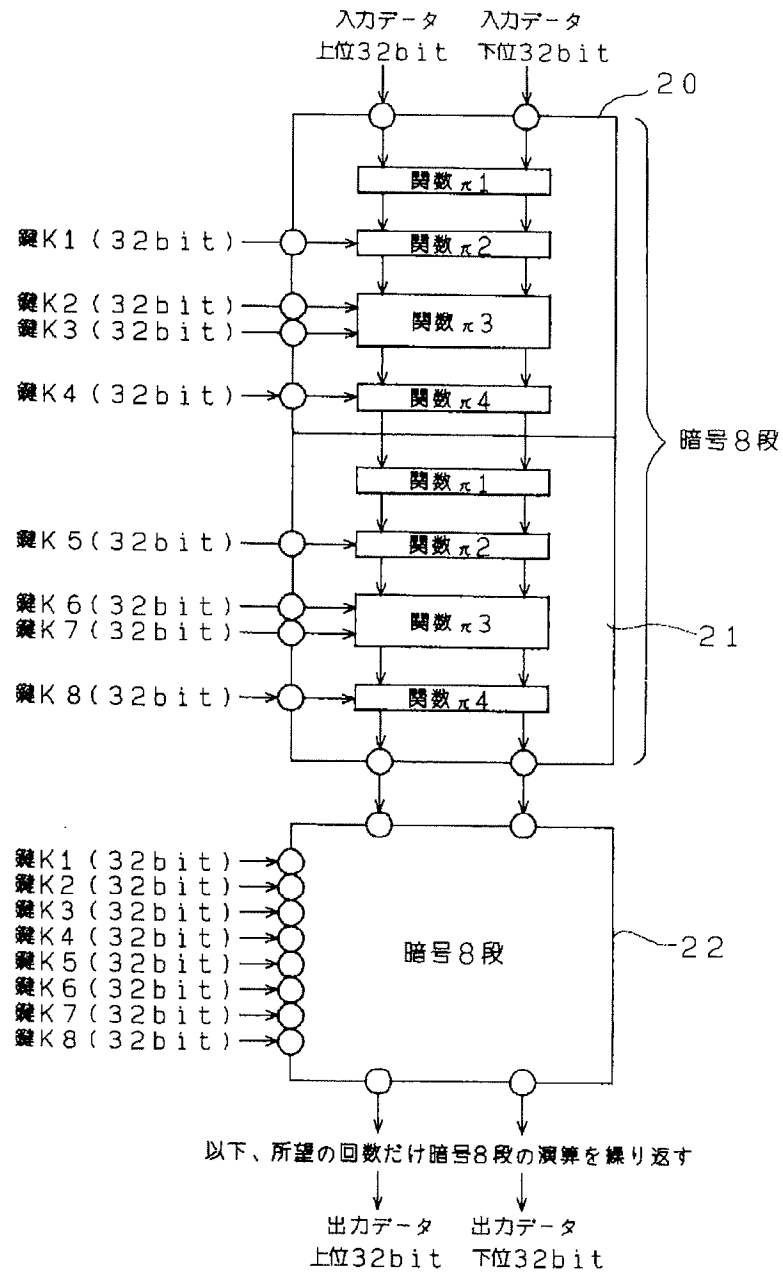
【図17】



【図10】

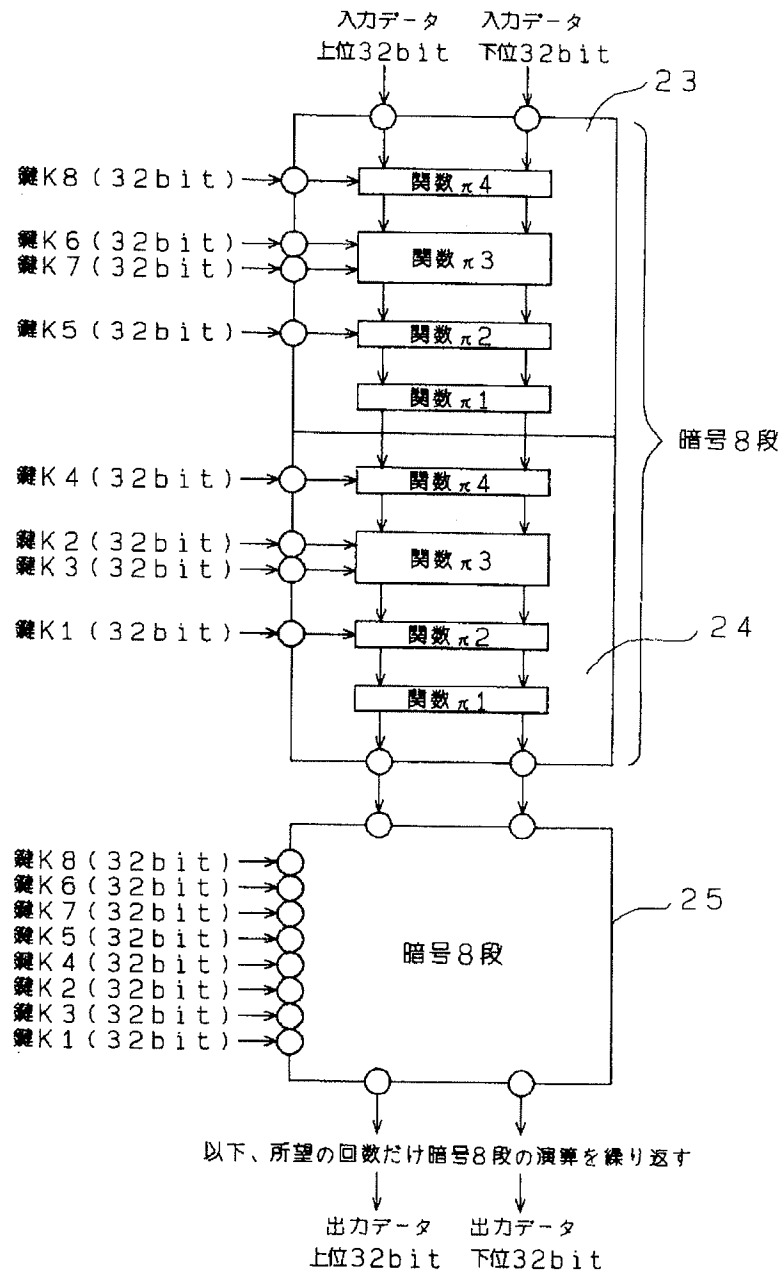


【図11】



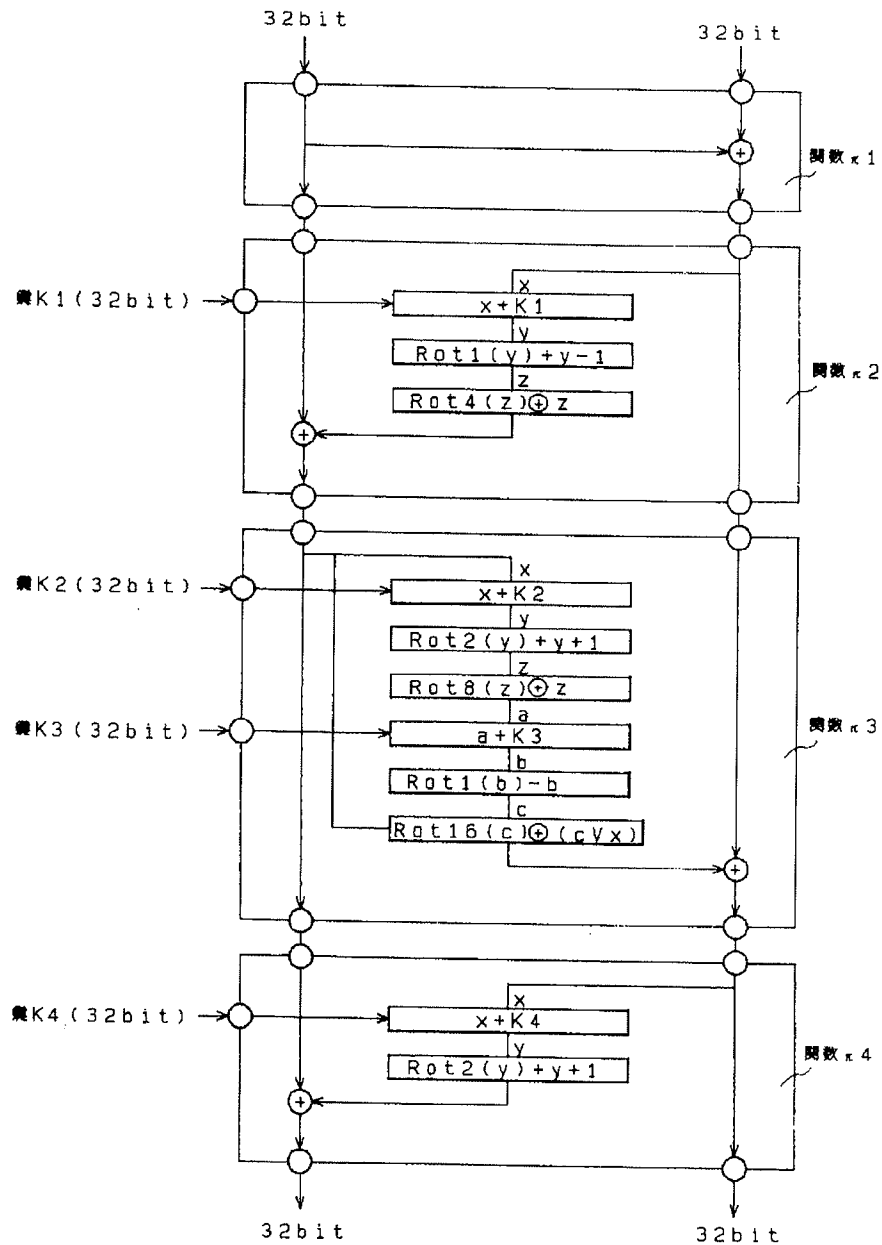
暗号処理

【図12】



解読処理

【图 13】



( 设备 )



:ビット毎の排他的論理和

+ :  $2^{32}$  を法とした加算

- :  $2^{32}$ を法とした減算

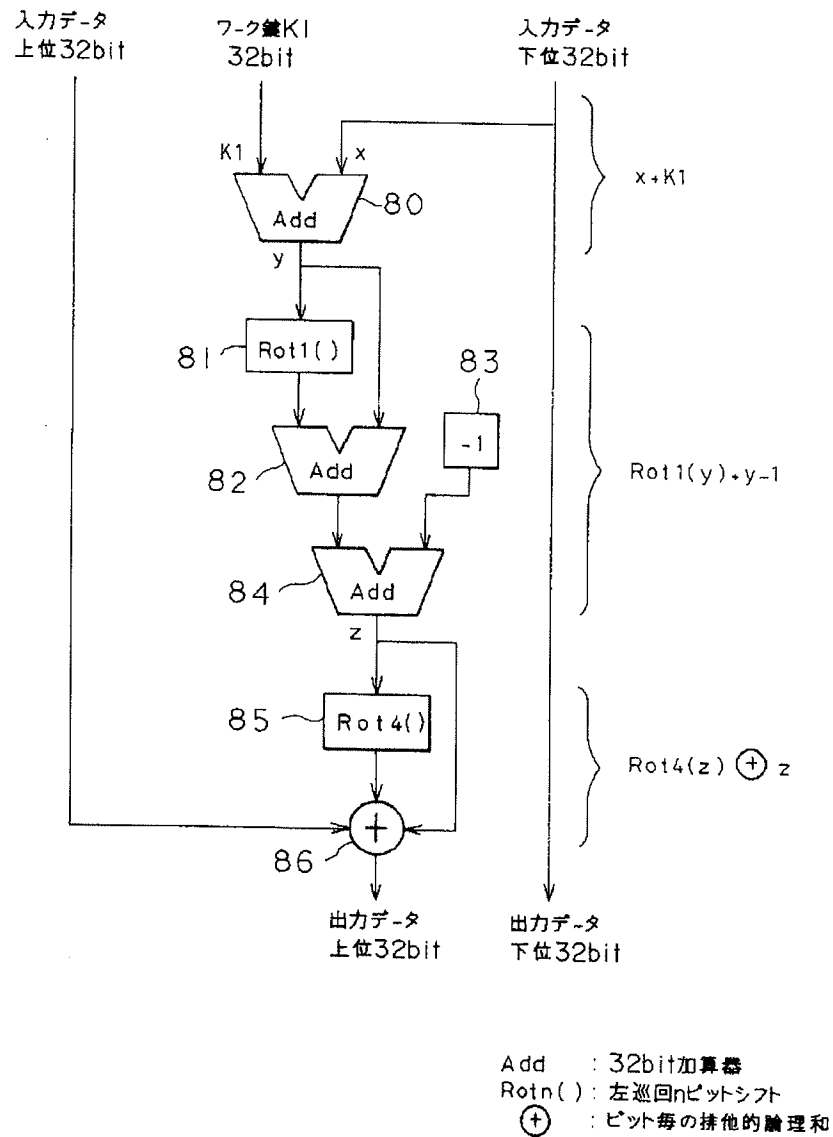
Rotn ( ) : 左巡回nビットシフト

Y:ピット毎の調理和

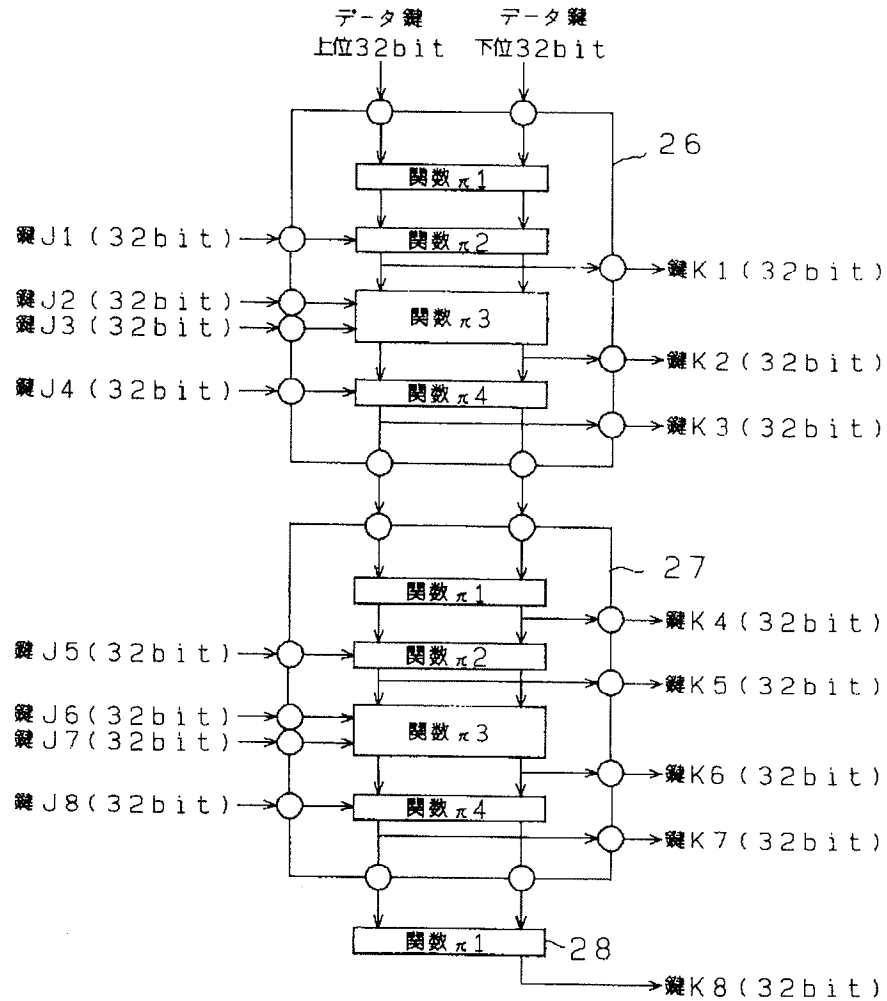
## 基本関数



【図14】

基本関数 $\pi_2$ の回路構成例

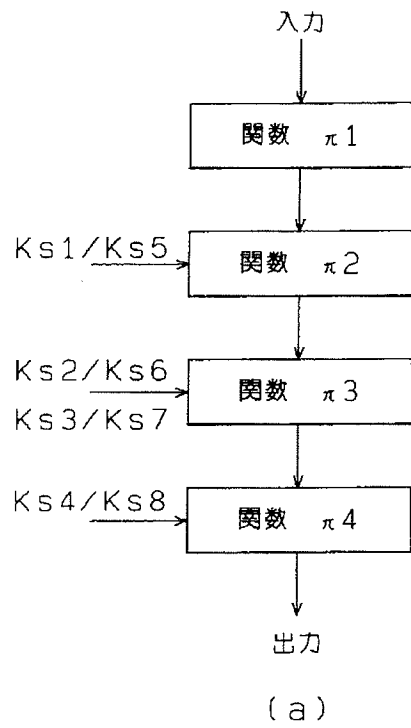
【図15】



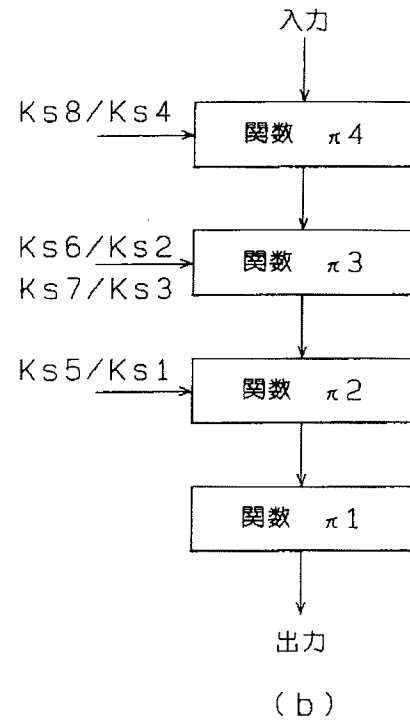
鍵スケジュール処理

【図16】

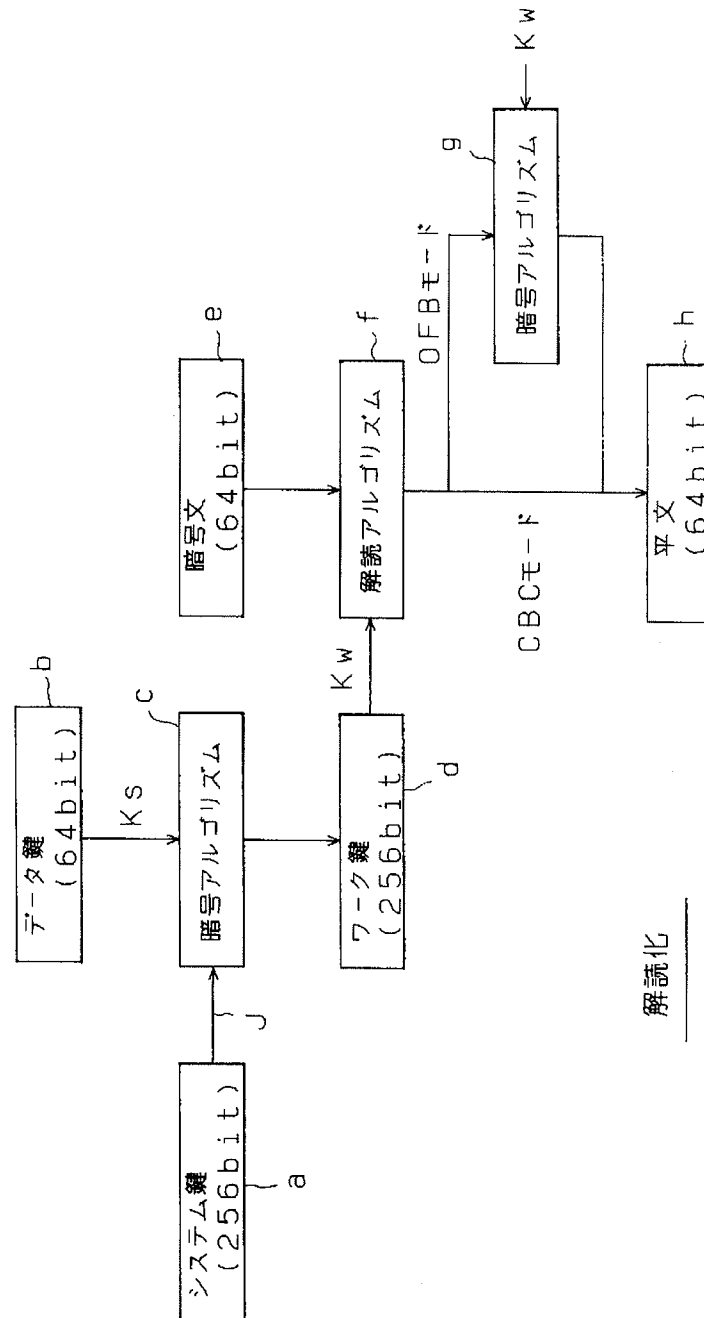
Encryptorコア



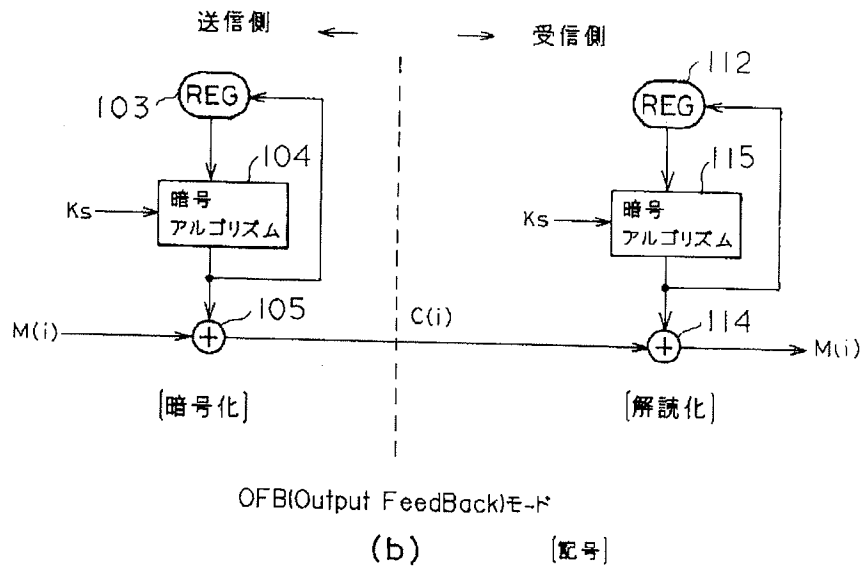
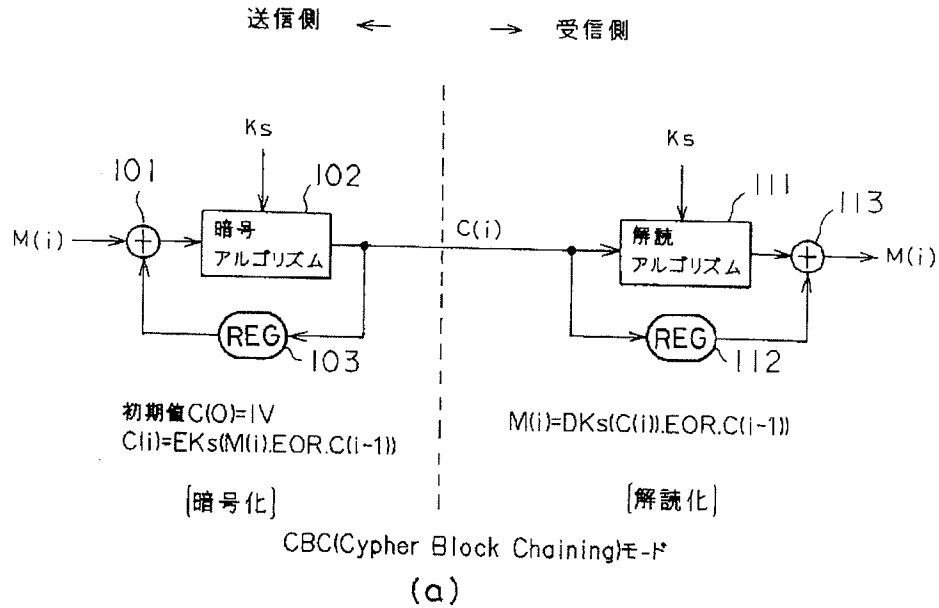
Decryptorコア



【図18】



【図19】



暗号化利用モード

$Ks$  : データ鍵  
 $M(i)$  :  $i$ 番目の平文ブロック  
 $C(i)$  :  $i$ 番目の暗号文ブロック  
 $IV$  : スケランブル用の初期値  
 $EK(m)$  :  $k$ で $m$ を暗号化  
 $DK(c)$  :  $k$ で $c$ を解読化

【図20】

